Integrating Community with Collections in Educational Digital Libraries

Monika Akbar

Dissertation submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Clifford A. Shaffer, Chair
Edward A. Fox
Stephen H. Edwards
Weiguo Fan
Susan H. Rodger

December 9, 2013
Blacksburg, Virginia

Keywords: Educational digital library, Online community, Deduced social network, Content ranking, Recommendation system

Integrating Community with Collections in Educational Digital Libraries

By
Monika Akbar

(ABSTRACT)

Some classes of Internet users have specific information needs and specialized information-seeking behaviors. For example, educators who are designing a course might create a syllabus, recommend books, create lecture slides, and use tools as lecture aid. All of these resources are available online, but are scattered across a large number of websites. Collecting, linking, and presenting the disparate items related to a given course topic within a digital library will help educators in finding quality educational material.

Content quality is important for users. The results of popular search engines typically fail to reflect community input regarding quality of the content. To disseminate information related to the quality of available resources, users need a common place to meet and share their experiences. Online communities can support knowledge-sharing practices (e.g., reviews, ratings).

We focus on finding the information needs of educators and helping users to identify potentially useful resources within an educational digital library. This research builds upon the existing 5S digital library (DL) framework. We extend core DL services (e.g., index, search, browse) to include information from latent user groups. We propose a formal definition for the next generation of educational digital libraries. We extend one aspect of this definition to study methods that incorporate collective knowledge within the DL framework. We introduce the concept of deduced social network (DSN) - a network that uses navigation history to deduce connections that are prevalent in an educational digital library. Knowledge gained from the DSN can be used to tailor DL services so as to guide users through the vast information space of educational digital libraries. As our testing ground, we use the AlgoViz and Ensemble portals, both of which have large collections of educational resources and seek to support online communities. We developed two applications, ranking of search results and recommendation, that use the information derived from DSNs. The revised ranking system incorporates social trends into the system, whereas the recommendation system assigns users to a specific group for content recommendation. Both applications show enhanced performance when DSN-derived information is incorporated.

This work received support from the National Science Foundation under Grant Numbers DUE-0836940, DUE-0937863, and DUE-0840719.

Acknowledgments

It has been a blessed journey - exciting, humbling, and enriching in so many ways. I am thankful to have had the chance to work with two wonderful advisors. My heartfelt gratitude to Dr. Clifford A. Shaffer for his inspiration, enthusiasm, advice, and patience. I am equally grateful to Dr. Edward A. Fox for his guidance, patience, inspiration, and support. My life has been immensely shaped by their knowledge, wisdom, philosophy, and ethics, and I cannot thank them enough.

I thank other committee members Dr. Weiguo Fan, Dr. Stephen Edwards, and Dr. Susan Rodger for their valuable feedback from time to time. I would like to specially thank Dr. Susan Rodger for driving to Blacksburg twice for my preliminary proposal defense and the final defense.

I thank Alexander J. D. Alon for being a supportive colleague and a good friend. I thank Yinlin Chen, Eric Fouh, and Michael Stewart — with whom I shared great times developing AlgoViz and Ensemble. I thank the extended Ensemble family for their advice and insight.

I thank my parents for their love, blessings, and support throughout my life, my brother Roni for his encouragements. I thank Shahriar, my husband, for his unconditional support all through this journey. I thank Nabhan, our baby, for allowing us to be a part of his wonderful world. I thank my mother-in-law for visiting us after Nabhan was born and keeping us sane and serene during those times.

I also thank all the wonderful people in DLRL and the CSA department. Lastly, I thank Virginia Tech and the Blacksburg community for making my stay here a memorable one.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Digital Libraries (DLs) are a well-known solution for collecting and storing digital objects. These libraries usually provide a number of services including indexing, searching, and browsing. Domain-specific digital libraries such as educational digital libraries (edu-DLs) provide a gateway to educational resources. An edu-DL can aid in gathering, organizing, and providing access to diverse educational materials that are available online. These libraries usually provide the core services of a typical DL, including indexing, browsing, and searching [51]. To be useful, an edu-DL must provide access to a range of educational resources (e.g., curricula, book reviews, collections of teaching aids), and provide a wide range of services for the life cycle of information collection, creation, dissemination, use, and reuse. Many edu-DLs harvest resources from different data providers who host the educational resources. Thus, edu-DLs often index objects of diverse nature and act as a portal to the actual websites hosting the resources.

The abundance of educational resources in an edu-DL provides opportunities but also creates problems for users when searching for high quality material. Without information on the quality of resources it becomes difficult to locate usable resources from hundreds, if not thousands, of choices. Ideally the user community provides feedback in various forms such as ratings, reviews, comments, etc. that can be helpful to gauge the quality of the resources. One such edu-DL is the AlgoViz Portal (`http://algoviz.org`) which attempts to combine an educational DL with online community.

Similar to AlgoViz, other edu-DLs also host resources or metadata coming from distributed sources. There is often little to no information on the quality or usefulness of the resources, nor do the resources receive sufficient user feedback. In such scenarios, social navigation, used in many domains to harness collective behavior of the users, can prove to be useful [87, 33]. One way to gain information on users' behavior is to devise mechanisms that would allow an edu-DL to capture, store, present, and incorporate various usage trends for digital objects. Leveraging the collective knowledge of a community within an edu-DL not only can help users to detect common trends, but also can help them to effectively navigate through the vast information space to potentially useful resources.

Communities are an integral part of teaching. Educators are often part of an institution that supports professional networks. Research shows that teachers participating in professional communities gain knowledge and psychological support which helps them to focus on their goals and increase

their consistency and commitment towards their program [2]. Within an edu-DL, a community also can be beneficial by providing value-added content through community members' discussions, reviews, comments, and ratings of the educational resources. The factors that make it difficult to build and sustain online communities include lack of a suitable environment that fosters online communities [19]. Even when DLs provide community space, willingness to participate can decide the success of a community. A lack of active user participation in communities is prominent in many domains. While users of edu-DLs can play a critical role by providing their feedback and ratings on the content, not many choose to do so.

1.1 Research approach

This research focuses on finding the information needs of educators and identifying a set of functionalities that are important for the next generation of edu-DLs. We define online communities within edu-DLs and provide a rubric to evaluate such communities. However, for cases where there is a lack of active community even though the DL sees significant user traffic, we propose a mechanism to identify latent user groups. Even a portal built for a specific user community (in our case, education) must still support disparate groups of users. We can identify such groups by linking users via shared resources (e.g., a co-author network). Latent groups also can be formed based on user interest in different types of resources or topics. Information on latent user groups can be used to steer users to potentially useful resources according to others' behavior within the DL. We study methods to recognize collective knowledge within the DL framework by analyzing latent connections between users and user interests.

We propose the concept of a deduced social network (DSN), which is derived from user activities within an edu-DL. In the absence of adequate explicit user feedback we use implicit usage data to deduce connections between users. These deduced connections form a network similar to social networks where two users are connected via shared attributes. The power of deduced social networks lies in the fact that they are *deduced*. If we want to find user interest in particular pages we can *deduce* connections based on the pageviews. If we want to find user trends on downloading resources, we can *deduce* connections based on download patterns. Such connections also can be *deduced* among resources (e.g., pages). Analysis of these networks and their contextual information can reveal interesting user behavior, different user roles, and communities with similar interests. Knowledge gained from these analyses can be used to tailor DL services so as to increase content accessibility.

Various DL services can be enriched by the knowledge derived from a DSN. Browsing and searching are two core DL services that present content to the user by ranking that content. Often a user browses a list of entries sorted according to some criteria (e.g., alphabetic, pageview, average rating). Search results also are ranked. Factors that can affect the ranking of search results include title of the page, number of pageviews, content of the page, etc. Since many of these fields are generic, AlgoViz, for example, uses a custom ranking framework that favors AlgoViz-specific fields. However, this custom ranking lacks any usage information in part because such information (e.g., ratings, comments) are rare. We show that in the absence of explicit user feedback, implicit user data can be used to create DSNs that have the potential to provide information that can improve ranking. Compared to pageview, DSNs provide information on the diversity of the user group that viewed a

resource as well as weights connected with that diversity.

DSNs also can improve recommendations. Most current recommendation systems rely on active user participation (e.g., feedback, reviews, ratings, etc.). But these are most often lacking in an edu-DL. We show how passive user data (e.g., clicks, pageviews, times in pages, etc.) can be used instead. Even with a target audience that is mostly anonymous, we are able to identify groups of users with specific interests. We propose a DSN-based recommendation framework that is able to model the likelihood of viewing a pair of pages in a session. Experimental results show our model is able to yield high classifier accuracy and promising recommendation performance compared to models that rely solely on text similarity.

As test beds, we studied two particular edu-DLS: AlgoViz and Ensemble. The AlgoViz portal has comprehensive collections of educational resources related to algorithm visualizations. The Ensemble portal is a distributed edu-DL for computing educators. Both of these DLs contain significant amounts of educational resources, and receive significant levels of traffic, yet both lack active user participation. We show our techniques can be used to improve content ranking and content recommendation even when there is no significant user feedback. We believe our approach can be used in other DLs that have little explicit user participation but abundant implicit user activities.

1.2 Research questions and contributions

Our broader goal is to support the information needs of educators, identifying a set of functionalities that are important for edu-DLs, defining and evaluating online communities within an edu-DL, and improving search and recommendation in the absence of explicit community. To do so we address two research topics in this document, each consisting of a series of questions.

1. What are the major components of a next generation educational digital library (DL 2.0)? What are the significant differences between DL 1.0 and DL 2.0? What is a suitable analysis of one of the most important differences?

In order to answer this series of research question, we studied edu-DLs, conducted focus groups to identify current resource-seeking trends of educators, analyzed their responses, and summarized our findings through the formal definition of the next generation of educational DLs. We analyzed one aspect of this definition in detail: online community within edu-DLs. We propose a definition for online community within edu-DL 2.0 and present a rubric for evaluating such communities. We also present four case studies that include analyses of the levels of community within the chosen edu-DLs.

We propose a mechanism to capture meaningful connections between users of an edu-DL 2.0. These connections can be represented using graphs. We refer to such a graph as a deduced social network (DSN). As an example, we use log data to create one kind of DSN that connects users via the webpages they viewed. A number of other features of an educational DL can be used to generate a network that has the potential to reveal interesting information related to user behavior. This leads to our second research question.

2. How can deduced social networks improve the performance of DL services such as ranking search results and recommendation?

In order to show the potential of the DSN concept, we built a number of DSNs using AlgoViz log data. Then we ran various analyses to identify and understand user trends. The knowledge gained from the DSN can be used to improve various DL services. We present two such applications - ranking and recommendation - that use the knowledge gained from the DSNs and exhibit improved performance. We also present a case study with another educational DL (Ensemble). Table 1.1 relates the topics with the chapters that address them.

Table 1.1: Research questions and chapter organization.

Type	**Topic**	**Chapter**
Theory	Current resource-seeking trends in edu-DL	Chapter 3
	Formal definition of Edu-DL 2.0	Chapter 3
	Online community within edu-DL	Chapter 3
	Rubric for online community within edu-DL	Chapter 3
Application	Building deduced social networks (DSNs)	Chapter 4, Chapter 7
	Ranking search results using DSN	Chapter 5, Chapter 7
	Recommendation using DSN	Chapter 6, Chapter 7

1.3 Dissertation organization

We present a literature review in Chapter 2. Chapter 3 describes our approach and findings on educational resource-seeking trends by educators – based on which we propose the next generation of educational DL. We also define the term "online community" and provide a community rubric followed by four case studies in this chapter. We propose the concept of deduced social network (DSN) in Chapter 4. We describe how an edu-DL can be represented using graphs and present various analyses on the DSNs. The findings from the DSN can be used to improve various services within an edu-DL. In Chapter 5 we use the knowledge derived from the DSNs to improve the performance of ranking of search results. Chapter 6 shows how DSNs can be used to provide recommendation in the absence of registered users' activity. We present a case study for the Ensemble portal in Chapter 7. Finally, Chapter 8 presents a brief summary of the tasks accomplished so far, along with future research directions.

Chapter 2

Related Work

In this chapter we present prior research that is related to different phases of our work. We start with work on digital libraries followed by social aspects of DLs. We then describe related work on online communities. We discuss relevant work on rank ordering and recommendation in the last two sections.

2.1 Digital libraries

Digital libraries host collections including digital objects and metadata of various scales, and provide services to access and use the collections. DLs offer various services related to the life cycle of information that includes collecting, organizing, archiving, preserving, and providing access to intellectual properties. During the earlier days of digital libraries a number of projects were supported by government agencies. The first phase of the Digital Library Initiative (DLI), funded jointly by the National Science Foundation (NSF), the Defense Advanced Research Projects Agency (DARPA), and the National Aeronautics and Space Administration (NASA), supported six projects over five years [55, 130, 129, 53]. The focus of the initiative was to understand how digital libraries can collect and store diverse electronic resources as well as provide access to them. The success of DLI Phase I led to further funding for Phase II, which explored the performance, scalability, and sustainability of past, present, and future collections.

There also have been efforts to define a digital library (DL) and its activities [52, 74, 20, 49, 22]. The DELOS reference model [20] proposes that a digital library universe consists of six major components that include content, users, functionality, quality, policy, and architecture. Content refers to the data and information stored within the DL; users are the actors — both human and machine — who interact with the digital library; functionality refers to the services offered by a DL; quality is a characteristic associated with the content and behavior of the DL; policy refers to various rules, regulations, terms, and conditions governing the DL; and architecture encompasses the overall DL framework. The DELOS model describes the digital library universe as a three-tier system with digital library (DL), digital library system (DLS), and digital library management system (DLMS). DLMS provides software and infrastructure to build a DLS that provides functionality and support

for DL activities such as collection, storage, management, and preservation of digital objects.

Goncalves et al. [49] proposed the 5S model where a DL is composed of Streams, Structures, Spaces, Scenarios, and Societies (5S). Stream refers to the properties of the digital object (e.g., text, audio, image), Structure indicates the organizational scheme of the digital objects (e.g., ontology, tag, link), Spaces refers to logical and representational views of the objects (e.g., vector space, index, user interface), Scenario refers to the services and activities supported by the DL, and Society refers to various communities and users of a DL.

Applications and systems also are developed to assist in building and maintaining DLs of diverse natures [75, 137]. DSpace, managed by the DSpace Foundation, is an open-source software system that supports building digital repositories [35]. Its ease of installation and deployment is one of the reasons why it is widely used in building digital libraries. The digital library research group at Cornell developed Flexible Extensible Digital Object and Repository Architecture (Fedora), a widely used digital object repository management system [104, 105, 125].

Interoperability between different digital libraries is important for sharing resources across these libraries. Standards to share metadata such as from the Open Archives Initiative (OAI) help ensure interoperability [38]. According to OAI-PMH (Open Archives Initiative - Protocol for Metadata Harvesting), actors fall into one of two roles, the data provider and the service provider. A data provider has a collection of metadata that it shares using the protocol. A service provider can harvest the metadata through a series of requests for metadata (using PMH), sent to data providers. Service providers usually also provide a number of services (e.g., searching, browsing) from its local harvested collection. Open-source software packages, such as jOAI [133], allow repositories to expose their metadata, following PMH.

Evaluation of DLs has been an active research topic. Fuhr et al. proposed an evaluation scheme for assessing the quality of a digital library from four dimensions: data, system, user, and usage [44]. Others also proposed quality models for digital libraries [50, 119]. Many of these pointed out the importance of understanding the needs of the target audience. Xie [138] identified major areas that contribute to the success of a DL that include usability, quality of collection, service, and system performance. All of these are building blocks of a successful information system [32, 121].

2.2 Online communities

Online communities depend on user interaction to become active and stay useful. Girgensohn, et al. [46] identified three sociological design challenges for building a successful socio-technical site: encouraging user participation, fostering social interactions, and promoting visibility of people and their activities. Koh, et al. [68] noted that participation can be of two types: passive participation (i.e., viewing) and active participation (i.e., posting), and each of these activities depends on different stimuli which includes active leadership, offline interaction, content usefulness, and sound infrastructure. User participation in online communities has been studied in depth from various angles. Nov, et al. [97] studied various motivations for different types of participation for varying levels of membership in the community. Luford, et al. [83] studied the effect of showing both similarity and distinctness information about a member and the groups where he or she belongs as a

means for increasing online community participation. Beenen, et al. [13] did similar studies based on social theories. Millen, et al. [88] investigated factors such as design decisions, member selection, and facilitating stimulating discussion as means of engaging the members of an online community. Preece, et al. [109] studied community members to find out reasons behind lower participation rates of a particular group of less active users known as *lurkers*.

Recent research uses log data to create behavioral networks to predict user activities [40, 39, 69]. Some of these approaches depend on link prediction methods. Current educational portals contain metadata coming from several different sources (i.e., collections). They are often organized by their original collection, hence many of those entries are not linked to each other even if they cover similar educational topics — making it difficult to predict links accurately. This is in contrast to methods of deducing behavioral networks that depend on user ratings.

2.2.1 Social aspects of digital libraries

Early DL research pointed out the importance of understanding the needs of the target audience and of building online communities [64, 138, 16, 17]. Online communities depend on user interaction to become active and stay useful. Researchers have documented the various types of participation, and discovered factors that motivate users to actively participate in those communities [97, 68]. Social navigation methods are used to guide users in an unfamiliar information space, but these methods largely depend on previous user feedback or ratings [87, 33]. In cases where user feedback or ratings is scarce, rating-based systems can prove insufficient to derive useful information.

Researchers have pointed out different aspects of establishing an online community in a DL [16, 84, 61]. There has been significant research on DL design issues [10, 54], studying and analyzing the overall DL architecture [31, 132], and identifying the success factors of online communities [77, 79].

Passive user activities such as clicks are used to recommend content in these scenarios [81, 140, 21]. Sites like Amazon have a successful recommendation system [80], that is targeted for e-commerce and depends heavily on user feedback. For educational sites, domain-based recommendation systems for e-learning were explored in [5]. In cases where user activity is less, recommendation systems based on social patterns were proposed [43, 40].

2.3 Content ranking

Ranking resources according to given criteria is a problem in many domains including information retrieval [117], recommendation systems [3], finding shortest paths [11], product rating [65], etc. Ranking approaches in information retrieval can be grouped into two broad categories: query-dependent and query independent. Query-dependent ranking depends solely on the query terms. Examples of such ranking includes the Boolean model, vector space model [117], and probabilistic models [85, 113]. Query independent ranking depends on links within the documents as well as query terms. PageRank [18], HITS [67], OPIC [1], etc., fall under this category.

Boolean retrieval models use a bag-of-words approach and often return unordered lists of matched documents for a given query. A vector space model on the other hand represents a document by

a vector [117]. The vector space model is one of the most used models in information retrieval. Each document is considered as a vector containing a set of terms (keywords). Similarity between two items in the vector space depends on the statistics of their terms. Terms may be given a Boolean value indicating their absence or presence in the item. Aside from Boolean value, TF-IDF (Term Frequency, Inverse Document Frequency) [118] is one of the most popular measures of term weighting. TF-IDF is used to assign weight to keywords retrieved from a set of documents. It considers both the local and global context of the keyword. TF (Term Frequency) shows the weight of a term in a specific item while IDF (Inverse Document Frequency) computes the occurrence frequency of the term in the repository. Using TF-IDF, words that appear more frequently inside a document but rarely in different documents show increased weight.

Assigning weight to terms only shows their relative importance. To find similar documents based on these weighted terms various similarity measures can be used. Cosine [42], Dice [42], and Overlap [135] are a few of the most commonly used similarity measures. Among these, Cosine similarity measures the cosine of the angle between documents. Documents that do not share any common terms have an angle of 90° resulting in a cosine value of 0. Documents sharing all terms will in contrast have a cosine value of 1. One of the advantages of Cosine similarity is that it is not influenced by the length of the documents. During information retrieval the query also is represented by another independent vector. The score returned by the similarity measure is used to rank the documents.

Query independent ranking uses link structure within the repository. Links within the Web can be grouped into two categories: back links and forward links. When page p_1 contains a reference to a page p_2, p_1 is said to be the back link of p_2. Similarly, p_2 is said to be the forward link of p_1. The underlying assumption of query independent ranking is that the importance of a page is related to its back links. In other words, an important page will be referenced more compared to pages that are less important.

One of the early formulations of this concept is the Pagerank (PR) algorithm that depends on the hyper-reference structure of webpages [18]. This algorithm gives a PR score to a page based on how many other pages point to it (i.e., back links). A high PR score of a given page P indicates a large number of pages are referencing P. Pagerank has a few shortcomings however, including problems with storing large data structures, taking powers of large matrices, and a low PR score of potentially useful pages due to less referencing.

A number of algorithms have been proposed to mitigate such issues [67, 1]. Among them is the Hyperlink-Induced Topic Search (HITS) algorithm [67]. HITS uses link topology to assign a page with two values: authority and hub. A page with a greater authority value indicates that this page is referenced by a large number of pages. A higher hub value indicates a hub page which acts as a catalog and provides links to other pages.

2.4 Recommendation systems

Personalization can be of two types: content personalization which allows users to view or change the content according to their needs, and service personalization which allows the users to tailor

services (e.g., notifications) or developers to tune the services (e.g., show similar content). One example of service personalization is recommender systems, which use various methods to suggest content to users. Recommendation systems have engendered great interest in recent years. As content continues to build up in various organizations around the globe, searching may not be the most productive way to extract useful information. Recommendation systems come into play in this scenario by using information on the context and history of the system. These systems have proven to be useful for online retailers [80], digital libraries [60], news groups, personalization of search results [12], and other areas.

2.4.1 User profiling in recommendation systems

Recommendation systems depend on user input for processing recommendations. Usually the users of a system provide some feedback or rating on the items they used or found helpful. This data could be structured like a rating of "4 out of 5" or unstructured like an evaluation. While it is convenient to use structured data for such systems, sometimes it is valuable to supplement numerical ratings through analyses of evaluative texts like comments. Unstructured data, however, can become hard to process since one word can have different meanings in different contexts. Thus retrieving the exact sense of a piece of a review can become challenging. Pre-processing techniques like stop word elimination, tokenization, and stemming are often used to get the most relevant form of the words used in any text document. While these are useful, they often fail to capture the context of the document. Existence of a word does not guarantee that the review is in support of the topic [107].

Manual user profiling

Aside from using explicit user input, implicit measures are often used for strengthening a recommendation. One of the most used implicit user inputs is browsing history. Implicit data is mostly used to create user profiles. User profiling can benefit from information like searching history, download history, or purchase history. Many systems use manual information gathered from users to create user profiles along with machine learning techniques [107]. Manual information is gathered via user interfaces where users select their areas of interest from a pre-defined set of choices. Amazon lets the user insert such information as 'Favorites'. Since this requires user's time and effort, many users are reluctant to provide such data. An alternative to this is rule-based profiling where rules are used to recommend similar items or items that are usually linked together. In the case of Amazon, if someone is looking for books, then the sequels of books already bought by the person would get the first preference in the recommendation list.

Automated user profiling

Research has focused on learning the user profiles for various recommendation systems [106]. User profiles are usually built using: machine learning techniques like clustering, classification, and decision trees [107]. Although probabilistic models like Naive Bayes classifiers [36] are widely used in automatic text categorization, they have potential for building user profiles as well. To learn a user profile, Mooney et al. [90] successfully used a Bayesian learning algorithm. Promising results

were seen when Bayesian classifiers were used to improve a collaborative recommendation system in [114]. Though studies have shown Naive Bayes classifiers often have better performance than other machine learning techniques [34], much work has been done to improve their efficiency further. Laplace smoothing [29] and Good Turing smoothing [45] showed promising improvements over the standard Naive Bayes classifier.

Once a user profile is created, it is then used to rank items and generate recommendations. Based on the type of data used, recommendation systems can be grouped into three categories: content-based, collaborative, and hybrid recommendation systems [12].

2.4.2 Content-based systems

Content-based recommendation systems depend on the content of the item for drawing out suggestions [107]. Items are described with features, and features are used alongside user profiles to find similar items that might interest the user. Depending on the type of content, extraction of a feature can be difficult. For example, while measures like TF-IDF [118] and Information Gain are most popular for extracting features from text documents, it is not easy to extract features from audio, video or multi-media items [12].

Examples of content-based recommendation systems can be found in various fields like e-commerce [25], domain-based suggestion of e-learning material [5], multi-media content recommendation [91] etc. Agent-based personalization has been proposed to improve the performance of content-based systems [131]. Agent-based personalization works on client-side data to build a user profile, which provides another layer of filtering on the content-based recommendations.

Usually content-based systems suggest newly arrived un-rated items based on a user's ratings on some existing items. For an online community, in the beginning there is usually less content. Depending on the number of users and their rate of contribution, a content-based system can be ideal for the initial starting phase of an online community. During this time sufficient information on user preferences may not be available. Thus modeling user profiles would be difficult, and so more weight on the content is needed for suggesting an item.

2.4.3 Collaborative systems

The second type of recommendation system is collaborative systems. Diverse areas like document recommendation [24], social network analysis [71], and Usenet news [70] rely on collaborative systems to cope with information overload — finding useful resources in a reasonable amount of time when there is too much information to manage. In collaborative systems, similarity between users is calculated in order to find the ratings of users that most likely have similar preferences. Reviews and ratings of similar users are used to find the most favorable item that could be of possible interest. Thus topics of recommendation are not limited to similar subjects, and based on the population trend it can vary widely. One of the first examples of collaborative systems is Tapestry [48] which allowed users to help each other perform filtering over emails or electronic documents. This system also supported content-based filtering. Online retailers like Amazon and Netflix use collaborative systems to provide best-matched items for a particular user's taste.

While collaborative systems show some promising results, they are not entirely free from errors. If a user has limited ratings on which to select his peers then the selection procedure may generate the wrong peers. Even with the right peers, the size of the peer group might not be sufficient to gather all of the features needed for ranking. If an object is new and not yet rated by anyone in the peer group, it might get a low ranking or no ranking at all. One strategy to overcome this could be to use a classifier to assign new objects to a pre-defined class. If the peer preference points to a certain class, then all the items of that class will go through both a collaborative and a content-based filtering. This brings us to the idea of hybrid recommendation.

2.4.4 Hybrid systems

Hybrid recommendation systems use a combination of both content-based and collaborative recommendation systems. Fab, a web-based recommendation system [12], uses both of these techniques to collect and rate items. The three main parts of Fab are the collection agent for finding specific topics, the selection agent for finding specific users, and the central router. Fab also requires the user to provide a rating of the results of recommendations, which then is used to update the user model of personal preferences. The ratings are used to generate recommendations of similar user profiles. MoRe, a hybrid movie recommendation system [78], uses two variations. The 'substitute' version depends on collaborative filtering but moves to content-based filtering when collaborative filtering cannot make any prediction. Another version, called 'switching', also uses collaborative filtering as its principal system but moves back to content-based filtering when the ratings gathered from collaborative filtering cannot pass a certain threshold. Yoda was developed to assist large-scale web-based applications that require accurate real-time recommendations [122]. Graph-based techniques were used in [60] to build a hybrid recommendation system for digital libraries.

Chapter 3

Educational Digital Library 2.0

We start this chapter by restating the first series of research questions, followed by a brief description of the approach we take to answer these questions. We then provide details on our findings in the following sections.

What are the major components of a next generation educational digital library (DL 2.0)? What are the significant differences between DL 1.0 and DL 2.0? What is a suitable analysis of one of the most important differences?

Our first step was to conduct focus groups to identify current resource-seeking trends in edu-DLs. Our findings identified some shortcomings of past and present edu-DLs. Based on our analysis we propose a set of features that are deemed necessary by the educators for future edu-DLs. We call the digital libraries that have these features "*edu-DL 2.0*".

There are a number of differences between the first generation of edu-DL and edu-DL 2.0. We provide a comparison between them in Table 3.3. One major area where edu-DL 2.0 is different from DL 1.0 is the inclusion of online community. We propose a definition for online community and provide a rubric to evaluate a community within an educational DL. We also present case studies in light of the definition and the rubric.

3.1 Current resource-seeking trends in educational digital libraries

Educational digital libraries serve as a gateway for finding educational resources online. In order to identify the current resource-seeking trends of the educators within edu-DLs, we conducted two focus groups with Ensemble[1], an edu-DL that seeks to support educators who teach computing [6].

The participants of the focus groups were faculty members of the Department of Business Information Technology (BIT) at Virginia Tech. This department has a unique pool of computing educators who teach IT and CS courses to Business majors. We invited 10 faculty, of which 9 were present for two different sessions, each an hour long. We followed a two-step process of data collection and

[1] `computingportal.org`

analysis, as described in Appendix D.

Our questions to participants were split across two broad topics: (i) How do they search for educational materials; and (ii) feedback on the Ensemble portal. We posed a set of 10 questions, listed below, to all participants.

1. How do you search for resources to use in a course, lesson or assignment related to an IS/IT related course?
2. In which content areas would you normally seek resources to support learning and teaching?
3. Which formats might be most helpful to your teaching or your students' learning?
4. Which resources do you have the most difficulty finding and accessing?
5. How do you stay up-to-date in your field in terms of education?
6. Which websites do you visit or which materials do you make regular use of? Why?
7. Do you use publisher sites often for your assessment needs?
8. Do you participate in any special interest groups (SIGs) or meetings to enrich your teaching or any social group? Do they have an online community site for it?
9. How valuable do you consider the use of badges and rewards in building an online community?
10. What are your thoughts about the Ensemble website?

We followed the grounded theory approach [126] to analyze the data. Initial coding was done to identify recurring themes or examples related to a theme, which resulted in around 30 codes. Many of these codes listed various areas of an underlying broader theme which helped us to identify different aspects of the code. For example, the code 'Ease of navigation' referred to various aspects of navigating through a site. While some participants argued that 'organization' of content is a major issue for easy navigation, others were inclined to better search mechanisms. Data analysis was done independently of the questions. Participants provided more information as we progressed through the sessions, causing the same code to be linked with multiple questions. At the end there were 246 references to these codes in the original transcripts.

The data indicate that educators often seek high quality resources, resources in specific formats (e.g., Powerpoint, PDF, video, animation), and resources that are customizable. They use Web search, university sites (e.g., MIT OpenCourseWare), publishers' sites, and personal connections to find the right resource. These users prefer websites with easy navigation, robust search, and a user-friendly interface. Educators who are interested in sharing their resources want contribution methods to be easy, want to be able to set up differential access to resources (e.g., assessment materials are only available to other educators), and want peer recognition for their contribution.

Figure 3.1 shows some of the top codes with their reference counts. For example, participants mentioned format or type of the resources a number of times. YouTube and educational video clips were mentioned as both motivating tools for students and informative resources. There were also

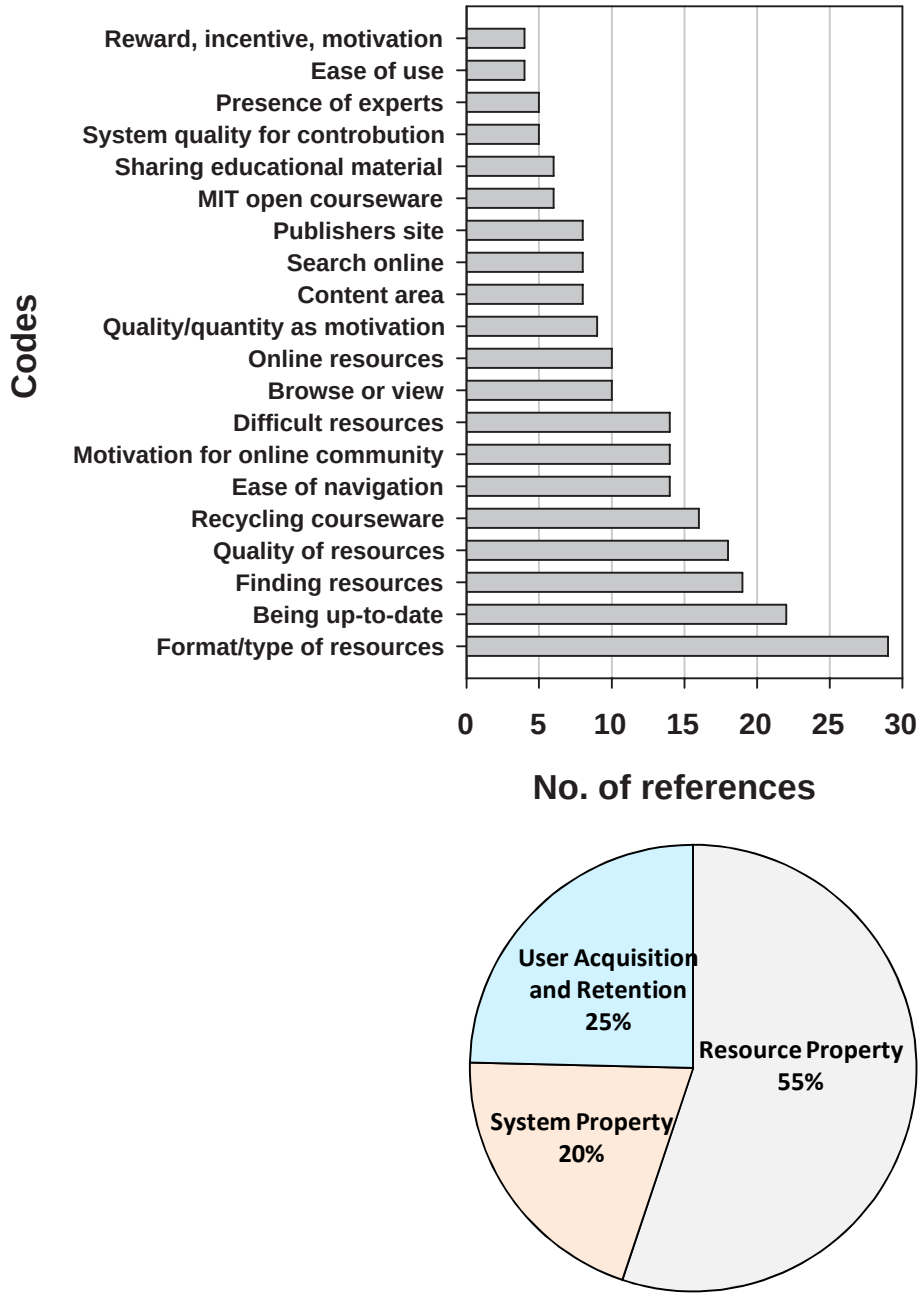


Figure 3.1: (top) Sample codes with reference count. (bottom) References distribution into themes.

mentions of syllabi, lecture notes, and PowerPoint slides that educators often seek in the Internet. Quality of available material was also a big concern (18 references). Many participants pointed out that they reuse or borrow existing course material as a starting point (16 references). Ease of navigation in an educational resource site also has big impact on the users (14 references).

Table 3.1: Emerging themes from the focus group data.

Resource Property
Format: Types/formats of educational materials.
Finding resources: Finding resources through Web search (e.g., Google), university sites (e.g., MIT OpenCourseWare), and personal connection.
Quality: Quality of available resources at various sites.
Recycling Courseware: Reusing course material or borrowing course content.
System Property
Factors influencing site usage.
Ease of Navigation - Organization of content: Easier topical organization following any standard organization scheme.
Robust Search: Visible search box/tab and granular searching options.
Interface: Takes less time to get used to the DL and use the resource.
Association between content (e.g., linked resources, taxonomy, ontology).
Factors influencing contribution.
Ease of contribution: Contribution should not take time.
Personalization
Notifications: Ability to subscribe to resources and users.
Content customization: Ability to customize textbook or assessments.
Add content to user list: Create personal collection from existing resources.
Differential access to resources: Access control to resources, especially for assessment materials.
User Acquisition and Retention
Motivation for using the site.
Existence of quality resource.
Existence of large quantity resource.
Existence of peer reviews.
Existence of experts in the community.
Critical Mass: Large user base.
Saving Time as a motivation for joining an educational DL.
Motivation for contribution
Peer recognition.
Quality of community and resources in the site.
Reward, incentive.
Academic recognition for contribution (e.g., promotion and tenure).
Building reputation (e.g., roles, badges) based on user activities.
Peer recognition.

After the initial coding, we grouped the themes based on their relevance to a set of broader themes. Three broad themes emerging from this level were: *Resource property*, *System property*, and *User acquisition and retention* (see Table 3.1). *Resource property* includes types of resources used by educators, difficult to find resources, methods on how they find resources online, etc. *System property* lists various aspects of a site that encourage participants on using the site. *User acquisition and retention* refers to factors that motivate users to actively use the site and participate in the site.

Some of the initial codes related to each of these themes are listed below them in Table 3.1.

The codes in Table 3.1 contain many of the characteristics that participants think define an ideal edu-DL. These are similar to those of Web 2.0 [100] which provides a dynamic environment for users by supporting sets of activities that promote social interactions, encourage user contribution, or capture and highlight collective knowledge. First generation edu-DLs mostly provided indexing and searching capabilities. These libraries often emphasized cataloging or hosting resources and providing browsing and search interfaces. Though some of these libraries allowed users to have an account on the site, the functionalities that came with the account were limited. Widespread support for collaborative tasks were lacking in Edu-DL 1.0. Examples of edu-DL 1.0 include CSTA Source[2] and DLESE[3].

CSTA Source is an online repository of teaching and learning materials for K-12 Computer Science education. Resources are listed under five levels (Level one through four and SI) which are called communities. Each community addresses different topics and may contain sub-communities. Users can register for an account and subscribe to resources to receive notifications. While the site allows commenting, this service does not seem widely used by users. CSTA lacks any space that allows open discussion or promotes collaborative environment (e.g., forum, blog).

DLESE is the Digital Library for Earth System Education. It hosts a wide range of educational resources that can be used in teaching and learning at various levels (e.g., K-12, College, Graduate). Resources are suggested by community members, which include educators, students, and scientists. The DLESE Reviewed Collection (DRC) lists a set of resources that are considered exemplary, but this list was not created with user feedback.

Both these systems host collections and provide the core DL services of indexing, searching, and browsing. While CSTA allows user account creation, the functionalities that come with the account are not sufficient to provide a personalized experience in the site. Although both sites use the term *community*, the term and its role is not clearly defined in either case.

Current resource seeking trends indicate that we need more than the core DL functionalities. The next generation of educational DLs should provide both personalized and collaborative experiences. The emerging themes from the focus group data include more details on the user expectations of edu-DL 2.0 (see Table 3.1). As we found, *quantity* of resource as well as the *quality* of resource and services is important in serving the educators. They also need a better way to manage the resources. Along with resources, social interactions are deemed important. Our findings lead us to the formal definition of next generation educational digital libraries. In the next section, we propose the Digital Library 2.0 for educational resources (edu-DL 2.0) that takes a user-centric approach by providing services to connect users and resources, and hosts online communities.

3.2 Edu-DL 2.0: Resources, Users, and Services

Our focus group sessions uncovered a series of unmet needs for educational resources, which include a digital library with rich resources, dynamic interactions between users and resources, and an active

[2] `http://csta.villanova.edu/`

[3] `http://www.dlese.org/library/index.jsp`

virtual community.

Two key entities of DL 2.0, *resource* and *user*, came up during our sessions with the participants who mentioned a number of ideal services of DL 2.0 which relate *resource* and *user*. Different connections between and among *resource* and *user* can create different relationships between these entities that can provide better exposure of resources and can eventually lead to better use of content. In some cases, these relationships even can yield new content. For example, services that connect a user with resources might allow the user to generate new content in the form of ratings or reviews.

As stated earlier, three core themes that emerged from our focus groups are resource property, system property, and user acquisition and retention. Under the theme *resource*, format, quantity, quality, and ability to customize the resources were important factors for educational DLs to be useful. As for *system property*, participants preferred a usable site that provides various levels of personalization. In the last theme, *user acquisition and retention*, presence of a user community, presence of experts in the community, various rewards and recognition for participation within the DL, etc. were seen as important factors for attracting new members and sustaining current ones. Based on the findings, we propose a definition of *edu-DL 2.0* that builds upon the 5S definition of the digital library [49] to describe the next generation of educational DLs. The formal 5S definition of digital library is stated as:

Definition 1 *A **digital library** is a 4-tuple* $(\mathcal{R}, Cat, Serv, Soc)$*, where*

- $\mathcal{R}$ *is a repository;*
- $Cat = \{DM_{C_1}, DM_{C_2}, ..., DM_{C_K}\}$ *is a set of metadata catalogs for all collections* $\{C_1, C_2, ..., C_K\}$ *in the repository;*
- *Serv is a set of services containing at least services for indexing, searching, and browsing;*
- *Soc is a society. [49]*

Our definition of edu-DL 2.0 is composed of three basic elements: resources, users, and services.

Definition 2 *An educational digital library 2.0, Edu-DL 2.0, is a 6-tuple (Resources, Users, Services, RSR, RSU, USU), where*

- $Resources(R)$ *in a educational DL are data or metadata objects and information that are collected, created, captured, generated, stored, and shared in the digital library;*
- $Users(U)$ *is the set of people in a educational DL who interact with the digital library, for example, educators, students, researchers, developers, policy makers, etc.;*
- $Services(S)$ *refer to the operations that allow interactions between and among resources and users;*
- RSR *represents the modeling and representation of resources within an edu-DL;*

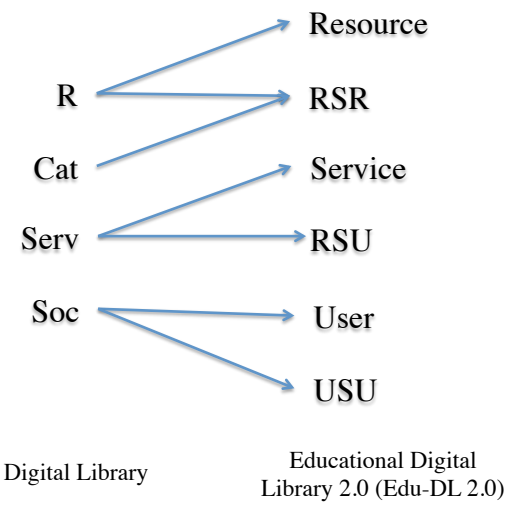
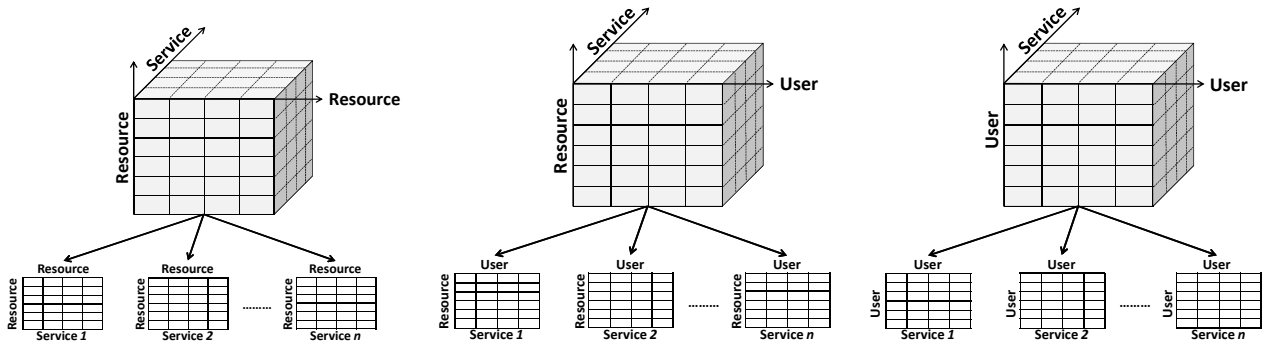


Figure 3.2: Mapping between the 5S definition of *Digital Library* [49] and *Edu-DL 2.0*.

- *RSU refers to the connections between users and resources indicating that there should be a number of ways to interact with the resources (e.g., comment, review, rate, tag); and*
- *USU refers to the connections users have with other users.*

Note that, in the trivial case, R and S also can refer to a single resource or service. Figure 3.2 shows a mapping between the two definitions. R and *cat* in the 5S definition of a digital library can be mapped to *resources* and various connections and structures present in those resources (*RSR*). Similarly, *serv* can be linked to *services* and connections that provide methods of interaction between users and resources (*RSR*, *RSU*, *USU*). Lastly, *soc* can be mapped to a single *user* as well as groups of users (*USU*).



(a) RSR Relational Matrix (b) RSU Relational Matrix (c) USU Relational Matrix

Figure 3.3: Relationship between *Resources*, *Users* and *Services*.

Table 3.2: Example of services (in **bold** text) for each relational matrix of Figure 3.3.

RSR	**RSU**	**USU**
• Linking resources (e.g., tags). • Associated resources (e.g., exercises linked to a lecture slide). • Peer reviews (e.g., ratings).	• A resource can have an **owner**. • A resource can be **read/downloaded**. • Users can **contribute** additional information (e.g., comments, ratings).	• Users can be **members** of a group. • Users can **contact** other users. • Users can be **connected via resources** (e.g., co-authors).

The three basic connections presented in the definition of Edu-DL 2.0 (i.e., RSR, RSU, USU) capture most of the preferred interactions between users and resources pointed out by the focus group participants. Figure 3.3 shows these three connections. This figure indicates that service is the connecting entity in relating resources with other resources (RSR), resources with users (RSU), and users with users (USU). Table 3.2 provides examples of some related services. Figure 3.3(a) shows the Resource-Service-Resource (RSR) relational matrix. For each layer in this relational matrix, there will be connection between two resources. For example, in a DL that allows users to provide feedback, a resource may have comments, which is another type of resource. Thus, two resources will be connected by the commenting service. Figure 3.3(b) shows the Resource-Service-User (RSU) relational matrix. A resource can be connected to users via a number of services that allow a user to be an author, reviewer, viewer, editor, etc. Figure 3.3(c) presents the User-Service-User (USU) relational matrix. This matrix captures the connections and interactions between users that would allow for a virtual social environment, which is desired by a large number of participants. Users can be connected directly to each other (e.g., member of a group, co-authors of a resource) or indirectly (e.g., viewed similar resource, rated similar resource, viewed resource with similar topic). The following subsections contain details on each of these connections.

3.2.1 Resource-Service-Resource (RSR)

More than half of the codes of our initial data analysis phase were related to some property of resources (see Figure 3.1(bottom)). *Organization* and *association* between resources are important to users. Participants identified a number of problems with various current organizational schemes used at different sites, with the most common problem being getting familiar with those different schemes. Every DL follows a different organizational structure, and to become familiar with a new navigational scheme is difficult. One suggestion was to use existing standards to create the categorization scheme. This would allow all resources to be organized by a set of well known topics. Use of non-standard terms also was confusing to many users. It was apparent from the discussions that some users always seek to understand the underlying organizational principle of a site, even when none exists. For example, in Ensemble we do not create any organizational scheme for the communities list. Though this is intended, according to one participant, the list seems like a 'hodge-podge' rather than an organized list.

Associations between content can be useful to users. Participants noted that they like to explore and use resources that are related to their course content (e.g., lecture notes linked to assessment materials). Multi-layered lesson plans at the NCTM Illuminations website[4] were appreciated by the educators. This highlights the fact that educators prefer different types of resources to be linked. DLs need to have a robust organization of content and proper associations between various resources.

Approachable *navigation* is important for encouraging users to explore a DL. Using deep navigation trees can be confusing. If the content is buried under five or six levels, a user often loses track of the context. Tags or lists with low depth can be useful. One suggestion was to show the context (e.g., tree, bread crumb). When applicable, information such as the link to the actual content should be 'eye-catching' or visually appealing. It was suggested that for a DL that hosts groups and communities, the navigation scheme should be consistent across collections, communities, and other sections.

Search is considered as an essential service for locating a resource. Several participants mentioned frequent use of advanced search features to locate relevant materials among a large number of resources. This feature is used even by those who are familiar with the site.

Quantity and quality of content is another frequently occurring code. Aside from the services on resources, more information on content also was noted as useful. Additional information can come in various forms. There can be descriptions of the resources, peer reviews, ratings, comments, or usage notes. All of this information requires that there be a 'group of users' who 'actively participate' in the DL.

3.2.2 Resource-Service-User (RSU)

One defining aspect of edu-DL 2.0 is that users play a key role. Static resources are not enough to meet many of the information needs of users, especially educators. There exists a need for a system that allows educators to interact with the resources and contribute easily. Systems that have peer reviews were appreciated by the participants. Such reviews can appear in various forms and require that a system is flexible enough to include services on the go, as needed. Above all, *ease of contribution* is critical to the success of edu-DL 2.0.

A prevalent practice among educators is *recycling courseware*. Depending on the audience and the syllabus, they may re-use some of the course materials or introduce new content. Thus, having the ability to *customize* the content to fit the demands of a course can be crucial to educators.

Usability is another issue for the next generation of DL. While users like more information, they also tend to prefer a clean interface. When the site contains much information, the *search* option is rapidly sought out by users. Getting used to the site should not take much time. As one participant explained, "*it is unlikely that someone would spend too much time to figure out how it can be used*". Time is a scarce resource for educators. They want a system that lowers their prep time, not one that requires time to understand.

One way that we can help users save time is by introducing *personalization* features such as an-

[4] `http://illuminations.nctm.org/`

notation or content tagging. *Notifications* can help users stay connected with the site. Several participants mentioned subscribing to newsfeeds. Being notified about chosen content or users is a form of personalization that can help the users stay connected while not taking too much time.

3.2.3 User-Service-User (USU)

Community feedback and peer reviews are important when trying to locate and use quality education material. Social interactions in virtual environments can take place in a number of formats including comments, ratings, and tags (CRT). Various sites depend on forums or blogs to share information on a larger scale. While most of these services create implicit connections between users, there are services that directly link one user with another (e.g., contact forms, message windows, groups). While these options would allow users to communicate with each other and stay connected, we first need to motivate users to visit the DL and explore the contents.

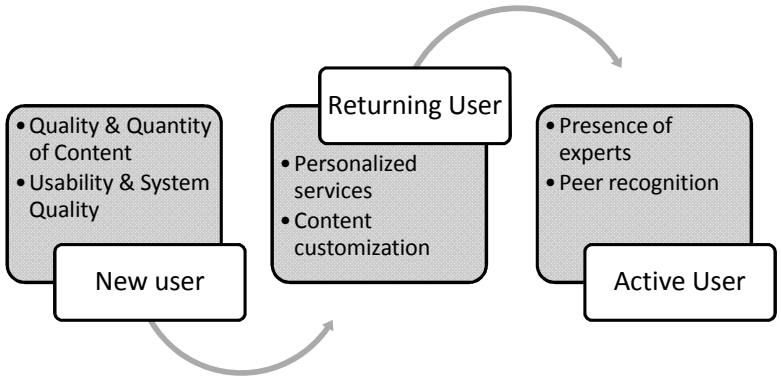


Figure 3.4: Types of users and motivating factors.

Various factors motivate users to visit an educational DL, use the materials, and actively participate in the community. Depending on the level of activity there can be different types of users. We broadly divide users into three categories: new user, returning user, and active user. Each type of users needs a certain kind of motivation to stay in that level or progress to the next level (see Figure 3.4). For new users, to be useful, a site has to be easy to get used to (*usability*), have quality materials (*resource quality*), provide useful services (e.g., advanced *search*, *notifications*). Motivations for returning users are a little different. Along with ease of use and high quality material, users also want the ability to create and share new content, customize content, or specify differential access to resources (e.g., assessments cannot be viewed by students). Returning users may start actively participating once they start getting used to the site and see value in contributing. Participants mentioned a number of incentives for motivating users to participate in the community. Of them, the *presence of experts* is crucial. *Active leadership* is also critical for a successful community. If contributions in the community are widely recognized as a valuable service, then they can be useful for career development. Both experts and novice users tend to value professional or academic recognition. Recognition can come in the form of *badges or rewards*. Sharing usage information with the contributor, which can be used as an impact factor, can be motivating to contributors.

3.3 Community in educational DL 2.0

Edu-DL 2.0 is the next-generation approach to educational DLs. Edu-DL 2.0 blends the traditional digital library contents with user-contributed content (ratings, comments, bookmarks, queries, etc.), and provides online community support (e.g., groups, blogs).

The core difference between edu-DL 1.0 and DL 2.0 lies in the fact that the latter is more dynamic, user-centric, encouraging user contribution, fostering virtual community, and incorporating knowledge with resources. While core services of traditional edu-DL 1.0 are limited to indexing, searching, and browsing, edu-DL 2.0 encompasses content management, dynamic services such as customization or personalization of content, and a collaborative environment. Table 3.3 provides a list of major differences between edu-DL 1.0 and 2.0.

One area where edu-DL 1.0 is different from edu-DL 2.0 is the presence of an active online community. Community feedback and peer reviews are important while trying to locate and use quality educational material. Such interactions in a virtual environment can take place in a number of formats including comments, ratings, and tags (CRT). Various sites depend on forums or blogs to share information on a larger scale. While most of these services create implicit connections between users, there are services that directly link one user with another (e.g., contact, message/chat, sub-

Table 3.3: Comparison between educational DL 1.0 and DL 2.0 based on the edu-DL 2.0 definition.

	Edu-DL 1.0	**Edu-DL 2.0**
Resources	Metadata, collection, repository, etc.	Metadata, collection, repository with user-contributed content (e.g., comments, ratings, reviews).
Users	Limited ability to serve individual user (e.g., user account, personalization).	Supports various tasks of individual user (e.g., registration, notification, rating, comment).
Services	Services include browsing, indexing, and searching.	Along with browsing, indexing, and searching edu-DL 2.0 provides personalized services, recommendations, user-friendly navigation, filtered search, etc.
RSR	Single listing of resources belonging to a particular collection/topic. Limited ability to annotate resource.	Cross-referenced resources across collections and attributes. Support for taxonomy, vocabulary, ontology, etc.
RSU	Limited ability for users to contribute in the DL.	Supports various levels of contribution and annotation from user (e.g., comment, rating, tag, sharing). Captures and utilizes user behavior in the DL.
USU	Does not explicitly support group-oriented tasks.	Supports groups, communities, and collaborations.

scribe a user). These options would allow users to communicate with each other and stay connected. It is not necessary that user activity within an online community be always explicit (e.g., comment, rating). Tracking implicit user activities such as pageviews, clicks, downloads, etc. also can show the presence of a community within a DL.

3.3.1 Online community within educational DL from different perspective

Communities also are identified as an important area of DL in the 5S DL definition [49]. The 5S definition lists *society* as one of its S elements. This definition of *Society*, presented next, includes conceptual communities.

Definition 3 *A* ***society*** *is a tuple (C, R), where*

1. $C = c_1, c_2, ..., c_n$ *is a set of conceptual communities, each community referring to a set of individuals of the same class or type (e.g., actors, service managers);*
2. $R = r_1, r_2, ..., r_m$ *is a set of relationships, each relationship being a tuple* $r_j = (e_j, i_j)$*, where* e_j *is a Cartesian product* $c_{k_1} \times c_{k_2} \times \cdots \times c_{k_{nj}}, 1 \leq k_1 < k_2 < \cdots < k_{nj} \leq n$*, which specifies the communities involved in the relationship, and* i_j *is an activity that describes the interactions or communications among individuals [49].*

Society in this definition is more general and can capture various entities such as hardware, software, admins, users. According to this definition, a society is formed by the activities between the entities of communities. The definition of edu-DL 2.0 (see Definition 2) encompasses the society aspect as the connection (i.e., *activities* in 5S) between users (*USU, RSU*). Services such as building groups, following another user, annotating a resource, etc., can create connections between users. Some of these connections may form a visible user community (e.g., group) while others can be used to deduce implicit user groups (e.g., co-author network). Based on the definition of edu-DL 2.0, we propose a formal definition of an online community within an educational DL 2.0.

Definition 4 *An* ***online community*** *within an educational DL is a 4-tuple (U, R, S, I) where*

1. *U is the set of users who share a common interest or goal,*
2. *R is a set of resources or events or activities that bring the members of U together,*
3. *S is a set of services, and*
4. *I is a function that connects the members of U via R or S.* $I : (R, S) \rightarrow U \times U$*.*

The term *community* is used broadly in Definition 3. According to this definition, communities may exist based on user roles or types. Though this definition captures the interactions within a *society*, it fails to define the activities that form a community. It is not clear if different types of users

can form a community, if any interaction between users is necessary to develop a community, or if communities can include sub-communities. Lastly, from a bigger DL perspective, it is not clear how a community might be connected to the DL resources. We extend this definition of community from the 5S definition of *Society* by defining how connections are formed within an online community in the context of edu-DLs (see Definition 4). Instead of limiting a community to a *set of individuals of the same class or type* we propose that a community may exist with users of different types or roles who are joined together by some shared interest in resources, events, or activities.

3.3.2 Evaluating online community within edu-DLs

Along with the focus group data, our study of the current and previous generation of edu-DLs revealed a set of characteristics that are deemed ideal or preferable for the success and usefulness of an online community. We summarize these characteristics in Table 3.4 as a rubric for evaluating a community within an edu-DL. This rubric is divided into two broad categories: Membership and Interaction. Membership includes a set of features related to users of an edu-DL. Interaction lists various levels of interaction between users and the resources within the DL.

Each of the broad categories in Table 3.4 has multiple sub-categories that provide more details. To begin with, an online community within an edu-DL 2.0 should offer rich membership services and allow various levels of interactions between and among users and resources. This is listed under the *membership* category which refers to various aspects of user accounts. *Membership* states that users should be able to register, create, and maintain their profile, choose different services such as notifications, and be rewarded for their participation. Membership also accounts for presence of experts and values active moderation within a community.

Interaction lists various activities that foster user interaction with resources and other users within an edu-DL. Users should be provided with collaborative environments such as groups, forums, and blogs. Contribution at various levels (e.g., comments, ratings) also should be allowed and rewarded. Usage information such as pageviews should be visible to users, thus providing a sense of underlying trends of resource usage.

Together, *membership* and *interaction* would provide a suitable environment for hosting online communities within edu-DLs. Many first generation edu-DLs are lacking in various areas of this rubric. Newer edu-DLs sometimes take advantage of content management systems that already address *membership* and *interaction* at various levels. In light of this rubric and Definition 2, we present case-studies of four educational DLs.

3.3.3 Case studies: online communities within educational DLs

As examples of edu-DL, we selected CSTA, DLESE, Ensemble, and AlgoViz. All these edu-DLs have core DL services such as browsing, indexing, and searching, which are cornerstones of the first generation of edu-DLs. However, some lack services that are important for edu-DL 2.0. Three out of the four edu-DLs emphasize computing education, while DLESE contains resources on earth science education. This provides us with a variety in subject matter for an edu-DL. The scope of the user base is also varied as we see in AlgoViz and Ensemble. While Ensemble aims to serve the

Table 3.4: Evaluation rubric for online community within an educational DL. Four edu-DLs are evaluated using this rubric.

		CSTA	DLESE	Ensemble	AlgoViz
Membership (U)	Ability to create and maintain user accounts	Yes	No	Yes	Yes
	Ability to contribute new content or annotate existing content (feedback, reviews, comments, ratings, tags, etc.)	Limited	Limited (comment)	Yes	Yes
	Value in user contribution (badge, ribbon, etc.)	No	No	Yes	Yes
	Manage account and content visibility	No	No	Yes	No
	Option to stay connected with the community (notification, subscription, RSS feed, etc.)	RSS feed	No	Yes	Yes
	Active leadership and presence of experts	No	Yes	Yes	Yes
	Active moderation	Yes	Yes	Yes	Yes
Interaction (I)	With members of U (create group, contact, chat, etc.)	No	No	Yes	Yes
	Level of interaction with members of U (closed group, open group, etc.)	No	No	Yes	No
	With resource R (create content, feedback, review, comment, rate, tag, etc.)	Limited	Limited	Yes	Yes
	Level of interaction with resource R (customize available content)	No	No	No	No
	Identify and share social trends. For example, share usage information with U (e.g., pageview, download, liked, shared, etc.)	No	No	Yes	Yes

computing education community in general, AlgoViz targets a smaller segment of that community that use visualizations in teaching/learning algorithms.

Computer Science Teachers Association - CSTA

CSTA supports and promotes computing education as well as the discipline at the national and international levels. This association was created to address various issues of computer science education at the pre-collage level including low enrollment, lack of diversity, gap between industry and academia, teaching and learning methods, etc.

CSTA provides resources on various areas of computing education such as curriculum, professional development, teacher certification, research, outreach, etc. The CSTA Web Repository[5] stores teaching and learning materials for computer science curricula for K-12. The scope of this case study is limited to this repository.

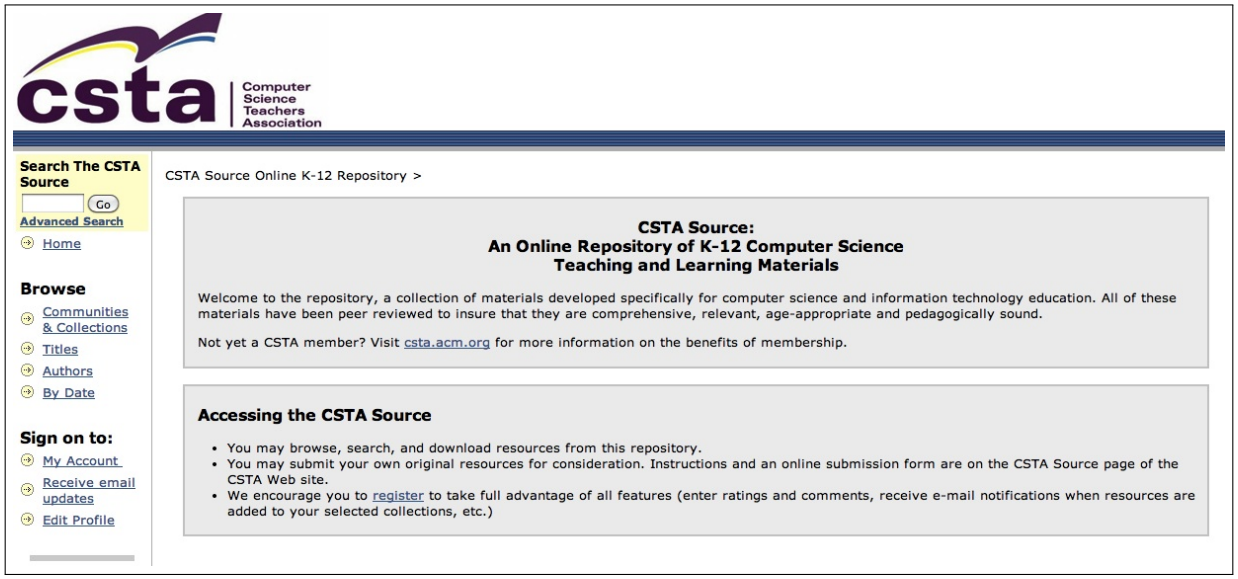


Figure 3.5: CSTA Web repository.

Users: The CSTA Web repository aims to serve CS educators by providing a repository of teaching materials. It allows users to create accounts and subscribe to content. Types of users who can find this repository helpful include, but are not limited to, teachers of computing education at various levels (elementary, middle school, high school, college/university), industry personnel, policy makers, curriculum developers, and researchers. Although the repository contains communities, the activities allowed by communities are limited to building and maintaining collections without any mechanism to allow or foster user interaction.

Resources: The resources in CSTA are organized around communities. Communities may form by administrative entries such as schools, departments, and research laboratories. A community can create an unlimited number of collections and is responsible for maintaining all of its collections.

[5] `http://csta.acm.org/WebRepository/WebRepository.html`

Each community has a top page with information on the collections, links for navigating (i.e., searching and browsing) through the collections, and information on recent submissions. Currently the resources are listed under five top-level communities within the CSTA Web repository: foundations of computer science (L1), computer science in the modern world (L2), computer science as analysis and design (L3), topics in computer science (L4), and strategies for implementation (SI). Many of these communities have multiple sub-communities.

Types of resources include posters, brochures, articles, lesson plans, source code, curricula, videos, etc. This repository also contains resources of various formats such as PDF, image, HTML, zip, Powerpoint, Word, etc.

Services: The CSTA Web repository allows users to browse the collections by titles, authors, subjects, and issue dates. The user also can search for resources. Advanced searching allows multiple keywords with logical operators (i.e., AND, OR, NOT). As the repository hosts the collections, users can download resources. Users also are able to submit resources, create an account, receive email updates, and edit their profile.

Membership: Users are able to create and maintain a user profile, subscribe to content, submit resources, and comment on existing resources. Provision for annotation of different granularity (e.g., rating, comment) are present. However, the repository fails to add any value in user contribution. There is low visibility of any active leadership and presence of experts.

Interaction: Users are not able to interact with other users. However, users can receive email notifications when new content is added to the communities they subscribed to. Ability to interact with resources also is limited to browsing, searching, and commenting. The repository fails to provide usage information of the resources, making it harder for users to gauge different aspects of resource usage such as popularity.

The CSTA Web repository is an example of the first generation of educational DLs with some advanced services. While core DL services such as indexing, browsing, and searching are present in this DL, it also facilitates limited user interaction with the DL through *membership* services which allow users to stay connected with the DL. However, factors that motivate users to engage in sharing their feedback (e.g., rewards, badges) are missing. Also missing is the presence of experts in the community. Any forms of interaction between users and resources are largely missing in this edu-DL. Thus it is not possible to build community spaces (e.g., groups, forums) or engage in anything similar. These factors make it less likely for an online community to grow and be sustained within CSTA.

Digital Library for Earth System Education - DLESE

Targeting a broader audience, DLESE[6] collects educational materials as well as Earth datasets and imagery. It aims to support Earth science education at various levels. It also provides information for developers to assist in building catalogs and collecting metadata following standard protocols.

Users: The repository is developed, reviewed, and maintained by a distributed community of educators, students, and scientists. It also provides a number of resources for developers who would

[6] `http://www.dlese.org/library/index.jsp`

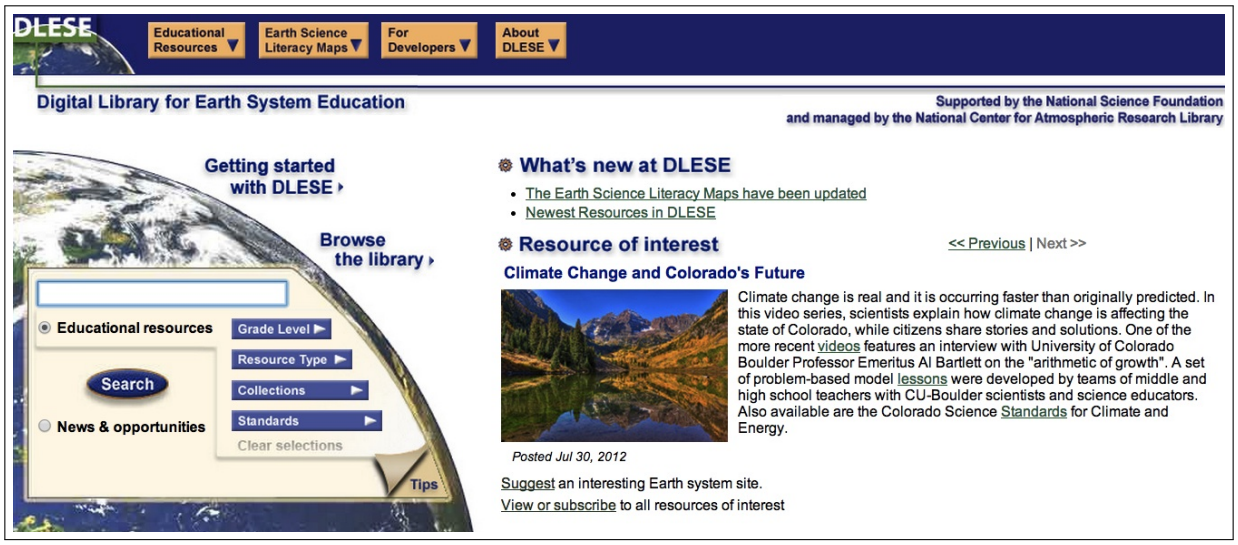


Figure 3.6: Digital library for earth science education.

like to connect to this library.

Resources: Resources are listed in three board categories: educational resources, earth science literacy maps, and developer resources. Educational resources include classroom and teaching material (e.g., classroom activity, computer activity, assessment, curriculum), visuals (e.g., image, video, visualization), text (e.g., article, book, report, proposal), audio, datasets, tools, etc. A set of literacy maps is also available for the educators. Developer resources include a set of APIs and information on metadata, collection building, and cataloging.

A wide array of various educational resources including lesson plans, visualizations, maps, images, assessment activities, and curricula make DLESE specially useful for both educators and learners.

Services: DLESE boosts a robust browsing schema. Educational resources are organized by subject (e.g., agricultural science, geographical sciences, geological sciences). It is also possible to browse this collection by grade level, which includes categories such as graduate/professional, informal, and general public. Two more browsing schema include browse by resource type and browse by standards. Users can suggest a resource and post news items. Search can be performed on news items or resources. The search results can be filtered using grade level, resource type, collections, and standards.

Membership: DLESE does not allow users to create accounts. Thus services such as user profile creation and maintenance, subscription, rating, commenting etc. are lacking in this DL. Users however can suggest a resource. The DL also fails to add any value in user contribution. Other than reviewing suggested resources before adding them to the core collection, the DL shows little, if any, proof of active leadership or presence of experts.

Interaction: Users are not able to interact with other users in any form. Ability to interact with resources is limited to browsing and searching. The repository also fails to provide usage information of the resources such as pageviews, downloads, etc.

The DLESE repository is an example of the first generation of educational DLs. It provides core DL services such as indexing, browsing, and searching. However it fails to provide most of the *membership* services in Table 3.4. It does not allow user account creation, provides limited ability to create new content (i.e., suggest new resource), and does not reward user contribution. Even staying connected with the DL is difficult since it fails to provide any service that allows notification of any form. Failing to recognize individual users and their contributions automatically makes it difficult to create and sustain any community within this DL. Thus the prospects of an active online community within DLESE is limited. DLESE also fails to provide most of the *interaction* services in Table 3.4. Users cannot interact with other users in any form (e.g., chat, forum, group) or any level. Interaction with resources is limited to comments. Usage information is not shared with the users, making it difficult for users to deduce the importance of a resource by following the social trend. Thus not only does DLESE lack in providing the users with a community space, it fails to share any underlying social trends as well.

Ensemble

Ensemble[7] is the NSDL Pathways project for computing education. It provides access to a broad range of existing educational resources for computing while preserving the collections and their associated curation processes. It also hosts community spaces and encourages user contribution, use, reuse, review, and evaluation of educational materials at multiple levels of granularity. It seeks to support the full range of computing education communities including computer science, computer engineering, software engineering, information science, information systems, and information technology as well as other areas of computing and informatics.

Users: Although targeted towards computing educators, the users of Ensemble include teachers, learners, researchers, policy makers, curriculum developers, etc. Ensemble allows user account creation as well as group formation. Groups can be open or closed. Various services allow users to stay connected to the site and interact with other users and resources.

Resources: Currently there are more than 20 collections of educational and informational resources on computing. Most collections are curated from data providers and their organization of content is preserved within Ensemble. Users can submit content which is added to Ensemble after peer review. The communities section lists a number of communities. Some of these communities have their own collection of resources. The technologies section lists a set of educational tools created and used to support computing education.

Services: Along with core DL services such as browsing, searching (faceted as well as federated), and indexing, Ensemble provides membership, subscription, and notification services to its users. It also allows users to submit new resources, as well as comment, rate, and tag existing resources.

Membership: Ensemble allows user profile creation and maintenance, subscription, rating, commenting etc. Users can suggest a resource to be added to the user collection. The DL adds value to user contribution by providing various badges for user activities. The DL shows the presence of experts in various communities it hosts. Active leadership also is maintained in those communities.

[7] `http://computingportal.org/`

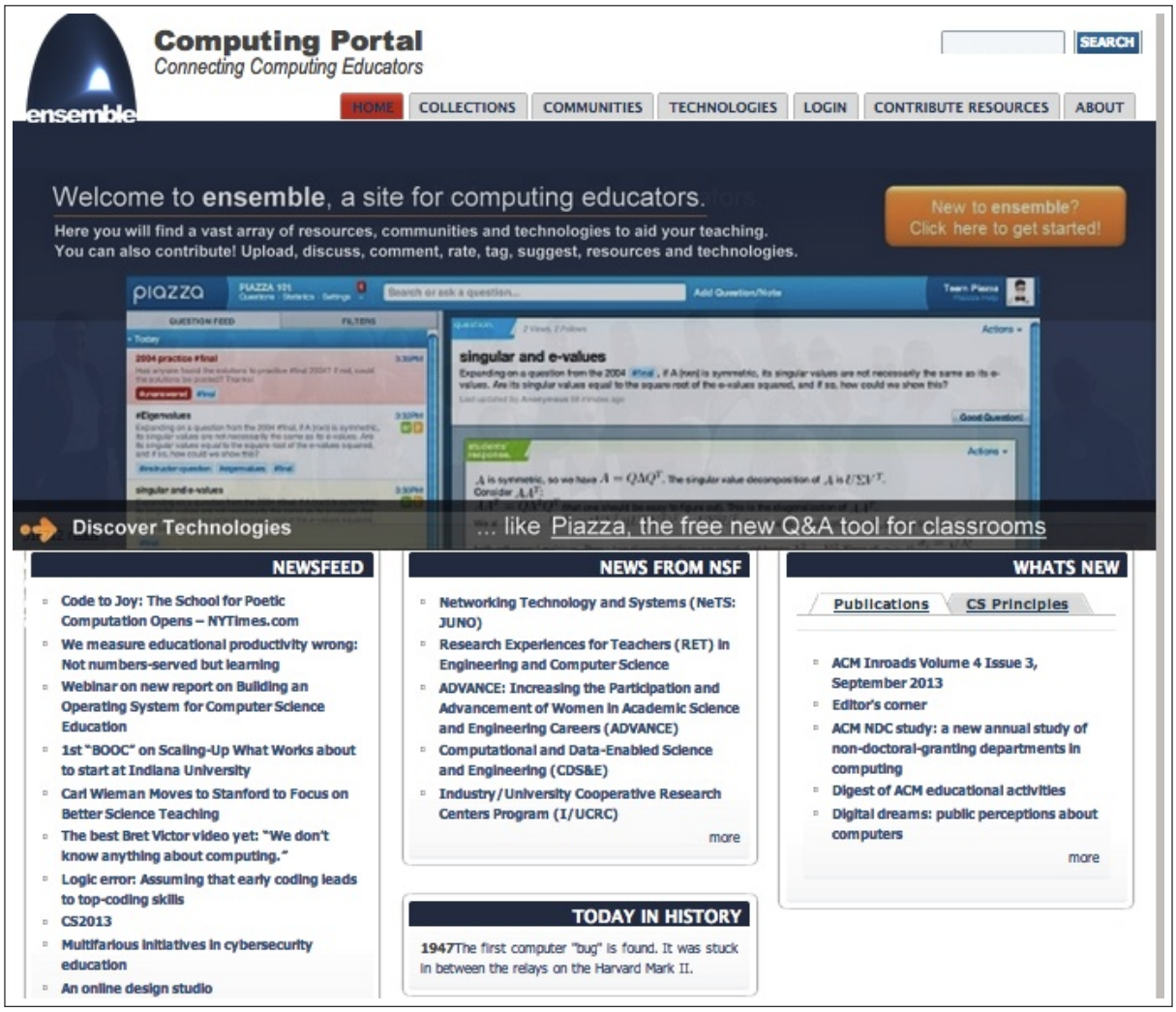


Figure 3.7: The Ensemble portal homepage.

Interaction: Users are able to interact with other users in many forms including message, subscription, forum, etc. Ability to interact with resources at various levels (e.g., rate, comment) is available. The portal provides usage information of the resources such as reads (e.g., pageviews).

The Ensemble portal is an example of edu-DL 2.0. It provides a wide array of *membership* services including user account creation, allowing user content contribution, placing value on such contribution with badges, providing various subscription services to stay connected to the DL, etc. Although it actively moderates the content, it fails to show any quality evaluation for the available resources. The presence of experts is limited to specific communities that are not readily visible.

Ensemble provides most of the *interaction* services that are deemed necessary for building and sustaining an online community. Users can communicate with other users in a number of ways (e.g., forum, group, contact message). Ability to create a community space (e.g., group) and manage its visibility (e.g., closed group) makes it particularly suitable for building an online community. This

is evident by the large number of communities it hosts (44 as of October 2013). Some of these communities contain more than a hundred users (e.g, CS2013, PACE). Many of these communities also contain their own forums, contents, blogs, etc. Ensemble provides a suitable environment for building and sustaining a community. However, these communities do not provide significant feedback on the educational resources. Ensemble can benefit from methods that can derive information on the quality of resources from other indirect measures. Ensemble also can aid users if the interaction levels between the user and resources are increased, such as users are able to build their own collection, edit existing resources, etc.

AlgoViz

AlgoViz[8] is a portal for algorithm visualizations. It collects, stores, and shares metadata on algorithm visualizations and animations (AVs). It is a gateway to AV-related services, collections, and resources. It also provides a research bibliography related to AVs.

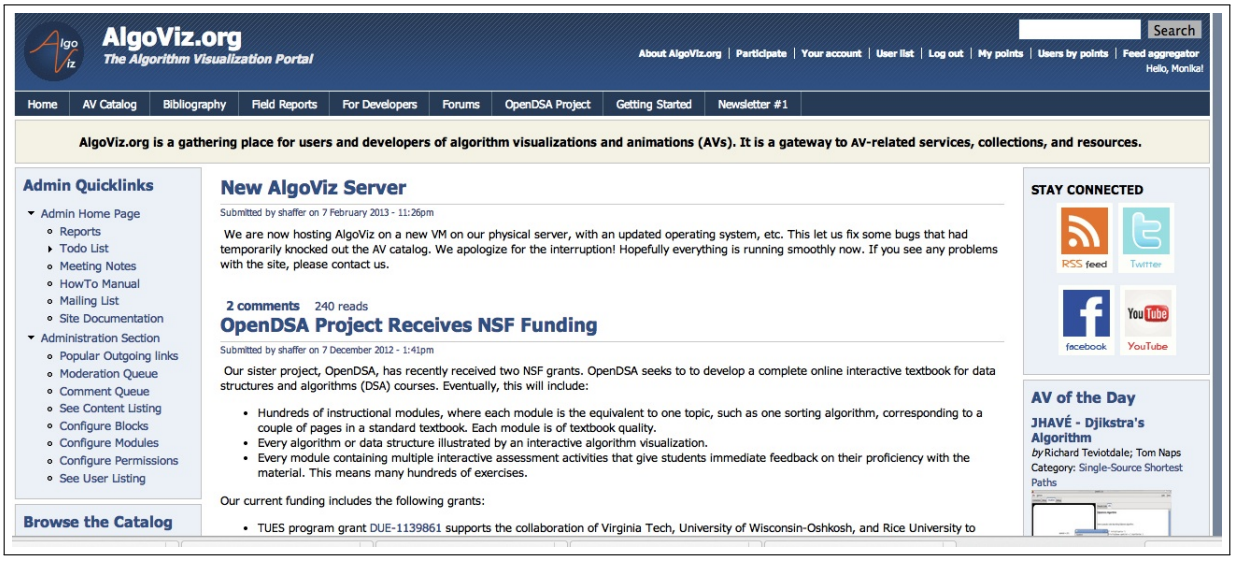


Figure 3.8: The AlgoViz portal homepage.

Users: The users of AlgoViz include teachers, learners, researchers, developers, and students. Users can register and create accounts, maintain profiles, and subscribe to content and authors to receive notifications. Although the resources are open for public viewing, any form of contribution requires that the user have an account with AlgoViz. Aside from forums, there is a section, called field reports, for educators to share their feedback on various AVs.

Resources: AlgoViz hosts a comprehensive catalog of algorithm visualizations. Entries in this catalog provide a detailed description of any given visualization including author and institute related to the AV, date of creation, language, format of the AV, topic it addresses, etc. There is also a bibliography of algorithm visualization related publications. Along with the forums, AlgoViz hosts a field reports

[8] `http://algoviz.org/`

section. News articles in the front page are used to share news on past and upcoming events related to AVs.

Services: AlgoViz includes a robust browsing and searching service that provides a number of facets to users to filter and narrow their searches. It also allows membership, subscription, notification, comments, review, ratings, etc.

Membership: AlgoViz does allow user account creation and maintenance, subscription, rating, commenting etc. Users can submit a resource which is reviewed by the admins. The portal adds value to user contribution by providing badges to active users. The portal also exhibits the presence of experts and active leadership.

Interaction: Users are able to interact with other users using the forum or contact form (once they are logged in). Ability to interact with resources at various granularity is also present (e.g., comment, rate). AlgoViz provides usage information of the resources such as pageviews. However it lacks services such as building groups.

The AlgoViz portal is an example of edu-DL 2.0. Although the user-base it serves is smaller and less diverse than Ensemble, both these edu-DLs have a lot of similar functionalities. AlgoViz allows extensive *membership* services that let users create and manage an account, subscribe to content, and stay connected with the DL through RSS. It further allows user content contribution, values user contribution (e.g., badges), and has active moderation. AlgoViz also allows the *interaction* between users and resources that is necessary for building online communities. Users can interact with others through contact forms or forums. However, the degree of interaction is low. It does not allow creation of groups or closed groups. Similar to Ensemble, personalization of resources is also lacking. Usage information is made visible to users. Overall, AlgoViz contains favorable services for building a community.

3.4 Summary

In this chapter we have presented data from two focus groups whose goal was to understand online information seeking trends of educators. Collected data indicate that educators desire to see improvements over existing digital libraries meant to serve them. Educators are in search of quality educational material and tend to borrow, adopt, or re-use those materials in their teaching, learning, and research. Based on these findings we proposed DL 2.0 services that tie together users and resources to create meaningful relationships. We described three necessary elements – *resources, users, services* – and three basic connections among them (RSR, RSU, USU) as the core elements of the next generation of educational DLs, i.e., edu-DL 2.0. We also define online communities within educational DLs along with a rubric for evaluating these communities. We concluded this chapter with four case studies. In the next chapter we will show how the interactions within an edu-DL 2.0 can be represented using graphs.

Chapter 4

Deduced Social Networks

Information overload is a problem for users of systems with large data collections. Navigating to useful resources can be difficult in such systems. Many educational DLs host hundreds if not thousands of resources. Researchers from different domains have devised solutions to information overload by modeling user behavior [82]. Many of these systems depend on user feedback along with the demographic information found in user accounts. This information can be used to model and predict user trends. However, edu-DLs often host collections with public access that users can navigate through without needing to create an account. Lack of user accounts poses a difficulty when building user models since these models depend on features and attributes derived from user accounts. Utilizing user activity to deduce latent user networks can help mitigate this problem in edu-DLs to some extent. Such deduced user networks can be used to improve DL services or introduce personalization.

Our research question for this and the remaining chapters is **How can deduced social networks improve the performance of DL services such as ranking search results and recommendation?** To answer this question, in this chapter we propose the concept of deduced social networks within an edu-DL. We then present a set of analyses done on DSNs. In later chapters we show how DSNs can be used to improve DL services.

Core services of a DL include indexing, browsing, and searching [51]. Educational digital libraries usually harvest large volumes of educational resources. An abundance of resources in educational DLs makes it harder for users, especially educators, to quickly locate useful content. Browsing and searching are two commonly used methods for finding content in a DL. We selected **ranking search results** and **recommendation** services to show the potential of deduced networks because search result ranking is an integral part of one of the core DL services (*search*), and *recommendation* is a service often found in edu-DL 2.0. We show that deduced user networks can enhance services that are native to both edu-DL 1.0 and 2.0.

This chapter describes various types of networks present in edu-DLs, provides a definition for deduced social network (DSN), presents case-studies for building DSNs in the AlgoViz and the Ensemble portals, and provides a number of analyses on those DSNs.

4.1 Networks in educational DLs

We saw in Chapter 3 that online communities are important for edu-DL 2.0. Yet building and sustaining an active online community is difficult [136, 83, 10]. Content in educational DLs mostly is free to use. The expectation, experience, and opinion related to educational resources vary from user to user. Even after a resource is used there is little motivation to provide feedback since evaluation activities take time and seem not to produce any tangible benefit for the user. This problem is made harder by the fact that the user base of most educational DLs is small compared to the large user base of e-commerce sites. In most cases, the users of an educational DL are not required to create an account. All of this makes it difficult to identify the users, their preferences, or usage trends within an educational DL.

While usage trends are extensively used by other online communities, those communities show certain traits that are missing or subtle in educational DLs. Those trends include a large user base, diversity in the user base, and mandatory user account/profile creation to access the service. Many successful online communities are e-commerce sites that actively encourage the buyer to share their experience and opinion of a product that they purchased.

One way to engage users in evaluation activities in a DL (e.g., view, rate, comment) could be to show what others have done. Systems that harness usage trends often provide cues to users that show what other users have done in a similar situation. Social networks can serve a number of purposes in a digital library. They can help in harnessing and spreading community knowledge and they can be used to identify common practices of users. They even can help users with a similar interest to interact with each other. Based on how social networks are created, we can divide them into two broad categories: **active** social networks and **passive** social networks.

Active Social Networks: Social networks that are actively created by the users fall under this category. There are websites that specifically allow users to link with other users by various methods including sending an acquaintance/friend request and accepting/rejecting that request. Social networking sites such as Linked In[1], CiteULike[2], or Delicious[3] follow this principle. Along with building their own network within these sites, users often provide information on their background, expertise, and preferences. Many of these sites allow users to form different groups or communities with specific interests.

Passive Social Networks: There exists another set of sites where the focus is more on providing information than on creating social networks. DBLP[4], a computer science bibliography library, is an example of such a site. It stores publication information which can be used to create co-author networks. Many digital libraries host resources that contain author information that can be used in similar ways. It also is possible to create networks based on user activities.

A user's activity within a DL can be grouped into two broad categories: **explicit** and **implicit**.

[1] `http://www.linkedin.com/`
[2] `http://www.citeulike.org/`
[3] `https://delicious.com/`
[4] `http://www.informatik.uni-trier.de/~ley/db/`

Explicit user activity includes the tasks that generate visible outcomes such as comments, ratings, feedback, etc. **Implicit** user activity comprises the tasks that are not visible to other users, such as pageviews, searching, browsing, downloads, etc. Repeated user activities that are similar in nature generate behavioral patterns. These patterns have the potential to lead users to resources that other users have explored. This chapter presents ways to detect and utilize behavioral patterns generated from implicit user activities.

Often, implicit user activities are recorded in user logs. Typical sites will have at least two log levels. The front-end log stores and shows usage information such as pageviews, number of comments, likes, etc. The back-end log is often used by system administrators to monitor and assess site performance. DLs also capture user activity in logs. This data can allow us to deduce underlying networks involving users. Such networks, which can connect one user with another based on their navigation history, can be interpreted as deduced social networks.

Passive or deduced social networks take an object-centric approach rather than the relationship-centric approach of active social networks. In the absence of active social networks, passive networks can help us understand how users interact with the resources in a DL. When used appropriately this information has the potential to improve DL services. A wide range of objects can be used in a DL as a basis to create passive social networks. For example, in an educational DL, there are often groups, collections of resources, and tools for teaching. It is possible to use these objects and user trends to form different networks. Here are examples of how a network can be deduced in such a DL.

1. We can connect users based on their activities in the site. For example, we can create a network of users based on their viewing of the same pages. In this network, users will be nodes and an edge between two users will indicate they have viewed some of the same pages. Later subsections will describe how to formalize and construct such networks.

2. It is not necessary that we only rely on user networks to find useful information. A connection between objects can be similarly interesting. For example, finding a group of objects that users view within a session can reveal useful information on their usage. Thus, pairs of users may be connected who are interested in objects from the same group of related objects.

A user base that generates explicit user activities (e.g., feedback, comment, etc.) makes it easier to identify user groups and usage patterns. However, while educational DLs experience a significant amount of traffic, they often lack explicit user activities in the form of reviews, ratings, comments, etc. In the absence of explicit user activity, we can depend on implicit user actions to identify such trends. We propose the concept of deduced social network (DSN), a social network of users and objects, which is independent of user profiles and entirely depends on log data to connect users based on their activities. DSN is a flexible concept such that it can be used to connect users as well as objects within educational DLs. Analysis of the DSN can lead to findings that have the potential to guide users through the information space of educational DLs.

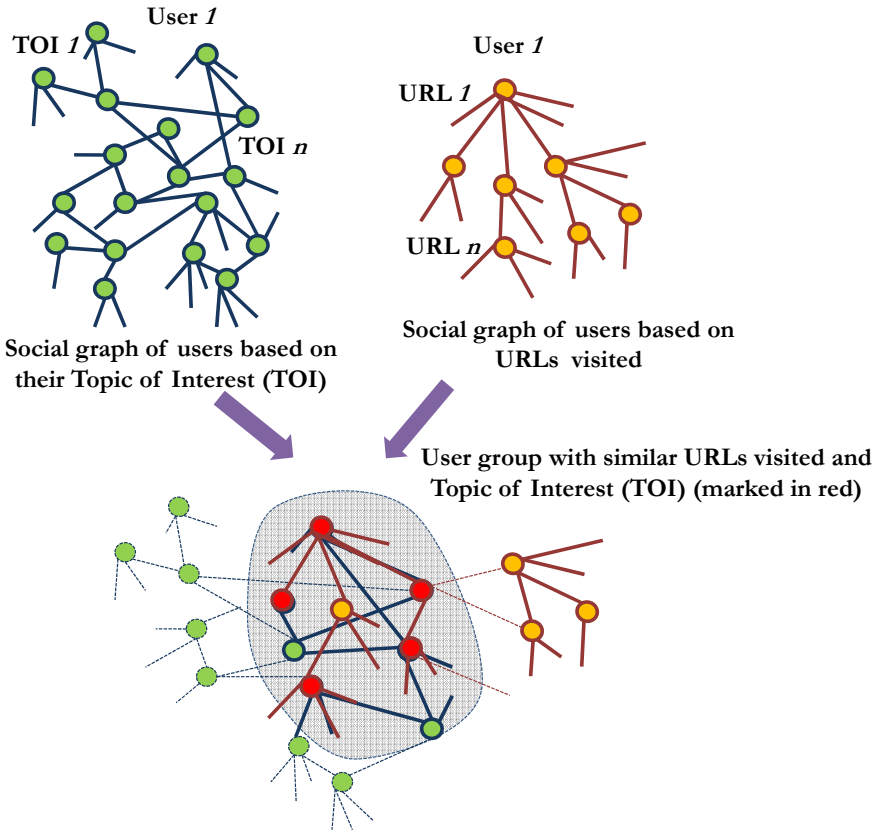


Figure 4.1: Possible social networks using different log information.

4.2 Formalization

Usage logs often show key pieces of information like average time spent on a certain page, bounce rate, and exit percentage for a webpage. Any of this information can be used to deduce connections between users resulting in deduced social networks (DSNs).

Figure 4.1 shows examples of deduced social networks [41, 7] and their potential to reveal interesting information. The top part of the figure shows two social networks constructed from information that might be available in the log data: topic of interest and URLs visited. The top left network with the green nodes connects users if their topic of interest is similar. The top right network is created based on the pages visited. The overlap of these two networks can reveal certain groups with special interests. The figure at the bottom shows such an overlap of two social networks constructed based on disparate criteria. The overlap results in red nodes, that show a group of users who share a topic of interest and have visited the same URLs. Thus, in the absence of an active social network, DSNs can help us identify groups of users with similar interests.

As seen from this example, the deduced social network is a graph that imposes thresholds on the edges.

We formally define a DSN as follows:

Definition 5 *A **Deduced Social Network (DSN)** is a graph with tuple (V, A, k), where:*

1. *V is a set of vertices,*
2. *A is the set of attributes of V which are used to create edges, and*
3. *k is a constant or a function that returns the minimum number of elements of A that must be common between two V to create a connection (i.e., edge) between them.*

In Figure 4.1, users are V; topic of interest and URLs visited are the attributes A that are used to create the top two networks; and $k >= 1$.

4.3 Detecting community with a deduced social network: AlgoViz

We now present a case-study for building and analyzing a DSN using AlgoViz log data. Figure 4.2 shows the architecture of the system we use in this case-study. The system consists of four segments: Filtering Module, Network Generation, Finding Communities, and Topic Modeling. We briefly describe these segments in the next subsections.

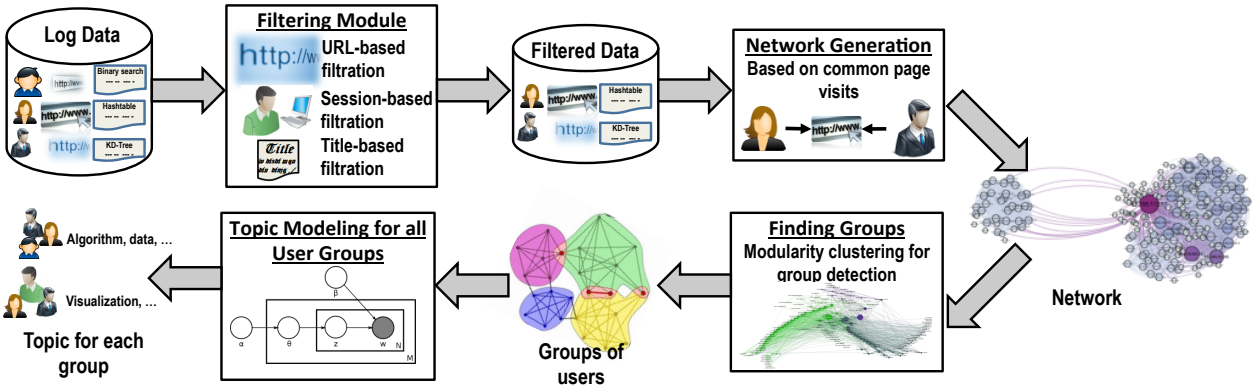


Figure 4.2: Architecture for community detection within a DL using implicit user data.

4.3.1 Filtering module

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, such log data includes activity by crawlers, spammers, and bots along with that of legitimate users. Thus the first step preceding further analysis is to filter the log data to remove non-user entries.

AlgoViz collects user history in several tables in its database. A sample of one of these tables, the Accesslog table, is given in Table 4.1. It shows data on the session, user ID, hostname, and

Table 4.1: Log data for AlgoViz.

Session ID	Page Title	Internal Path/Page URL	Hostname‡	User ID	Timestamp
ievav83	Lifting the hood of the computer...	node/1413	93.x.y.z	0	1276272047
t5fuuba	biblio/export/ tagged/118/ popup	research.cs. vt.edu/algoviz /biblio	207.x.y.z	0	1276260935
ivuks8s	Has an AV helped you learn a topic in computer science?	research.cs. vt.edu/algoviz /poll/	95.x.y.z	0	1276260943

‡ IPs are changed to protect user identity.

timestamp of when the page was visited. AlgoViz content is open for public viewing, hence it is possible for users to search for content without registering. These are referred to as Anonymous users and so have a default user ID of 0. We used hostnames (i.e., IP addresses) instead of user IDs to identify trends in our user base. For any given hostname, we are able to determine which pages were viewed in a session — identified by the *session ID* variable, and the time when the page loaded. We used these data to generate a deduced social network. The Accesslog table uses a variable named *access-id* (AID) as the primary key of the table. It also stores session information. Each page viewed in a session generates a new AID (i.e., row) in the table. Much of the data in this table are generated from spammers, crawlers, bots, etc. We followed a three-step process to clear the log data of non-user data.

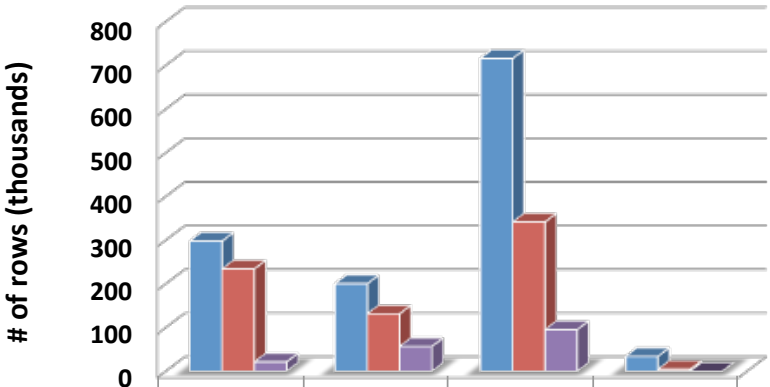
1. Filter data based on page titles: Many pages in AlgoViz are generic and less informative for understanding user behavior. At the first stage of data cleaning, we prune the rows based on the titles of the pages. Examples of pages titles that are less informative and less important include: 'Welcome', 'Access denied', 'Page not found', etc.

2. Filter data based on the path: Sometimes the title of the page alone is not sufficient to understand content. For example, the profile page for a user contains the user name as the title. The Accesslog table stores the internal path of the page. We used these paths to prune rows that include generic internal paths such as 'user/register', 'user/login', 'node', etc.

3. Filter data based on the session information: At this phase, we investigated user behavior based on the session information. In particular, we can detect aggregate behavior such as average pageviews per session, average length of session if at least two pages were viewed, etc. With such information, we are able to identify the outliers, possibly bots, in the log data, and filter them out. The session-based pruning was done in three stages:

   (a) Pages per session: Count of the number of unique pages generated in a session; prune if this value is greater than a certain threshold x.

   (b) Seconds per page: On average, in a session, how much time was spent on a page; prune if this value is less than a certain threshold y.



	Fall 2009	Spring 2010	Spring 2011	Spring 2013
# of rows after step 1	298091	200706	716095	34317
# of rows after step 2	234496	131793	341882	4841
# of rows after step 3	21656	57062	95599	253

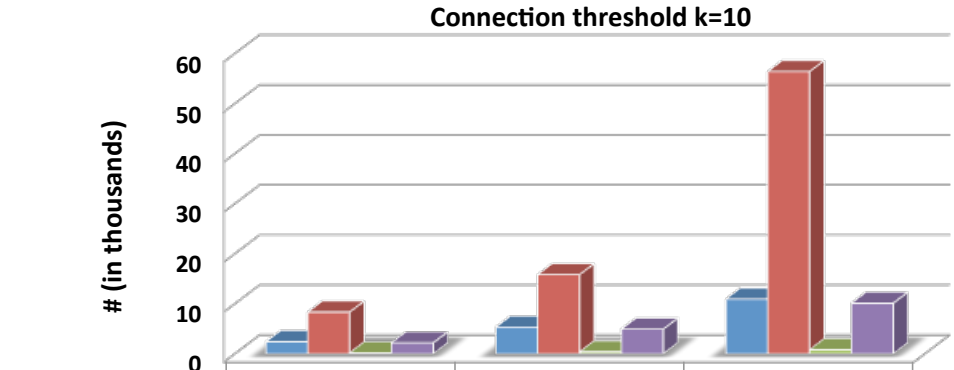
Figure 4.3: Data cleaning in AlgoViz.

(c) Number of sessions with one pageview: Usually a session would contain multiple pageviews. We observed a tendency to generate a large number of sessions with only one pageview, for certain hostnames. We pruned all of the rows containing each such suspicious hostname.

Figure 4.3 shows the number of rows in AlgoViz log data for four different semesters as the filtering module passes through the different data cleaning steps. The timeline for Fall semester is August 1st to December 31st, and Spring semester is January 1st to May 31st. Each row represents the loading of a page in a session. For example, if a user u viewed three pages in a session $sess_i$, this log will have three rows for user u with session $sess_i$. As we see in this figure, activities recorded in the log vary from time to time. While Fall 2009 and Spring 2010 have similar numbers of visits, Spring 2011 shows increased activities. Spring 2013 shows the least user activity of these semesters. For each semester, each cleaning step reduces the number of rows initially found in the log data. For example, in Fall 2009 there were 298,091 rows after the first data cleaning step, which become 21,656 after the last data cleaning step.

4.3.2 Network generation

Once the log data has been filtered, we can connect users based on their activities in the DL. The AlgoViz accesslog table has multiple attributes that can be used to connect pairs of users. Among these attributes we choose to use pageviews. That is, two users are connected if they viewed some of the same pages. These connections create the DSN. Each node in this network represents a user. We use a connection threshold parameter to vary the network strength. A connection threshold of size k for an edge indicates that two users viewed at least k common pages.



	Fall 2009	Spring 2010	Spring 2011
Nodes	2427	5352	10965
Edges	8358	15846	56432
Connected Nodes	203	413	837
Isolated Nodes	2224	4939	10128

Figure 4.4: Building the deduced social networks with AlgoViz log.

Figure 4.4 shows the number of nodes, edges, connected nodes, and isolated nodes in the DSNs for Fall 2009, Spring 2010, and Spring 2011. Isolated nodes are the nodes without any edge. We use a connection threshold of 10 for all these networks. This figure shows that in Fall 2009 there are 2,427 users. However, only 203 of these users have edges between them and the number of edges are 8,358. The number of users without any edge is 2,224. Among the 5,352 users in the Spring 2010 DSN only 413 were connected to each other. The total number of edges in Spring 2010 DSN is 15,846. Spring 2011 is the largest of the three DSNs with a total of 10,965 users and 56,432 edges. Only 837 users shared edges and the number of isolated nodes is 10,128.

Network attributes such as density, betweenness centrality, modularity, etc. change depending on the connection threshold. Density [27] for a network with edges E and vertices V is defined as:

$$density = \frac{2 * |E|}{|V| * (|V| - 1)} \tag{4.1}$$

Figure 4.5 shows the effect of varying connection threshold on the network in terms of network density. The X-axis shows the values of k and the Y-axis lists the density of the network. The three lines represent three DSNs. According to this plot, density decreases quickly as we increase k. For example, in Fall 2009, with $k = 6$ the density is 0.007 and with $k = 20$ the density is 0.0001. Similarly, in the Spring 2011 DSN, with $k = 6$ the density is 0.006 which drops to 0.002 as we change k to 8. Lowering the connection threshold adds more edges into the network, thus making the network more susceptible to outliers. On the other hand with a higher threshold, as the network starts to get sparse it may loose important information. We wanted to select a k that would generate a DSN in between these two extremes. The plots show that the density of the DSN changes rapidly until $k = 8$ and starts to stabilize around $k = 14$. We believe $8 < k < 14$ is a good

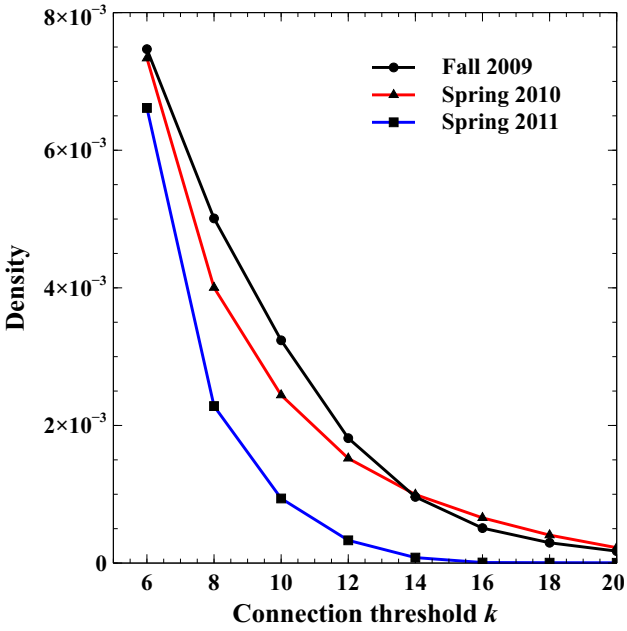
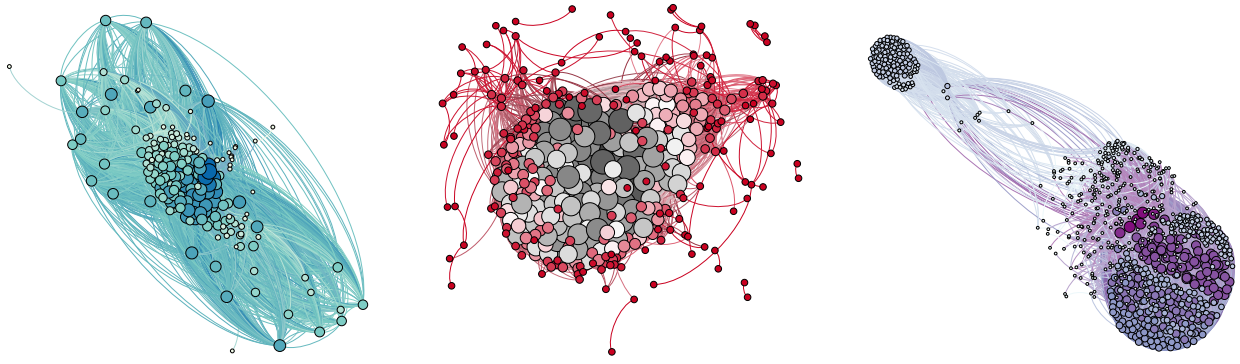


Figure 4.5: Effect of connection threshold in DSN.

range of threshold that would generate informative DSNs without much noise. For further analyses we opted to use $k = 10$ for AlgoViz DSNs.

The respective DSNs, without the isolated nodes, are shown in Figure 4.6. The size of the nodes in this figure are proportional to their degrees. As we see in these plots, various network attributes change depending on the timespan of the DSN and user activities during that time. For example, the Fall 2009 DSN shows largely interconnected nodes indicating most of the users were viewing similar sets of resources. However, in Spring 2010 we see many nodes with one or two edges. These are the users who viewed a set of resources where parts were similar to other users and parts were



(a) Fall 2009 (b) Spring 2010 (c) Spring 2011

Figure 4.6: Deduced social networks in AlgoViz.

unique. Finally, in the Spring 2011 DSN we see two clear groups of users even before partitioning the DSN. Although these groups are visible in the DSN, we need further analysis to identify the characteristics of these groups.

4.3.3 Network partitioning – finding communities

When it comes to understanding user trends, constructing networks alone is not sufficient. We need to study the properties of the networks in order to get meaningful insight into the users' behavior. Networks have been studied extensively in a variety of fields including physics, biology, and sociology [120, 101, 26]. Finding communities within networks is an active research area. Newman and Girvan used betweenness as a measure for removing edges and finding communities in graphs [47]. Clauset et al. used hierarchical agglomeration algorithms for detecting communities in large networks [26]. Among various clustering techniques, spectral clustering and modularity clustering also are gaining popularity in community detection.

Often network characteristics such as density, modularity, and connectivity are used to select the best algorithm for community detection. Some of the measures that are commonly used in community detection within a network include optimizing the minimum cut, hierarchical clustering, and modularity maximization. Many of these approaches depend on the idea of graph partitioning — breaking down the network 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 for Web science [62], epidemiology [56], sensor networks [116], and parallel computing [58, 73].

Modularity, introduced by Girvan and Newman, is a quality measure for clustering that has been successfully adopted in many areas [47, 94]. Modularity clustering is dependent on edge betweenness — a measure that assigns weight to an edge based on 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.

After we build a DSN, its characteristics might dictate the best methods to identify communities of users. Depending on factors such as k, the DSN can be dense or sparse, thus having different network characteristics (e.g., degree distribution, betweenness centrality). Factors that can influence and dictate network characteristics include the timespan of the log data used to define the DSN, the number of common attributes that creates a link between two entities, the nature of content within a DL, etc. These factors have the potential to generate sparse graphs, thus making it possible to readily identify different communities. For denser graphs, detecting communities can be tricky. Network characteristics such as node degree distribution should be used to find the proper graph partitioning algorithm suitable for detecting groups within that particular type of network.

We compare the performance of five different algorithms for detecting communities over four datasets. The first two datasets, AlgoViz and Ensemble, are created from logs - AlgoViz from September 2010

Table 4.2: Properties of the datasets used with graph partitioning algorithms.

	# of Nodes	# of Edges
AlgoViz - September 2010	195	2255
Ensemble - March 2012	89	721
US airport	500	2980
Facebook-like small social network	1899	20296

and Ensemble from March 2012. The third dataset, US airport[5], contains a list of 500 airports in the USA. An edge exists between two airports if a flight was scheduled between them in 2002. The last dataset is from a small-scale social network site[6] similar to Facebook. It originated from an online community for students at University of California, Irvine. The dataset includes the users that sent or received at least one message (1,899). Table 4.2 shows the datasets and their network properties such as number of nodes and edges. For the datasets generated from logs we report the number of nodes and edges that appear in the cleaned data.

The first algorithm is Spinglass [111], which was developed for statistical physics. According to this method each node can have a different spin state and the interaction between two nodes dictates if the nodes will stay in the same spin state. A simulation is then run to find out the different states which are identified as communities.

The next algorithm, multi-level [15], uses multi-level optimization of modularity to find communities. It takes a hierarchical approach to detecting communities. Initially each node forms a separate community. At each iteration nodes are re-arranged into communities such that the modularity is increased. The iteration stops when it fails to increase the modularity significantly.

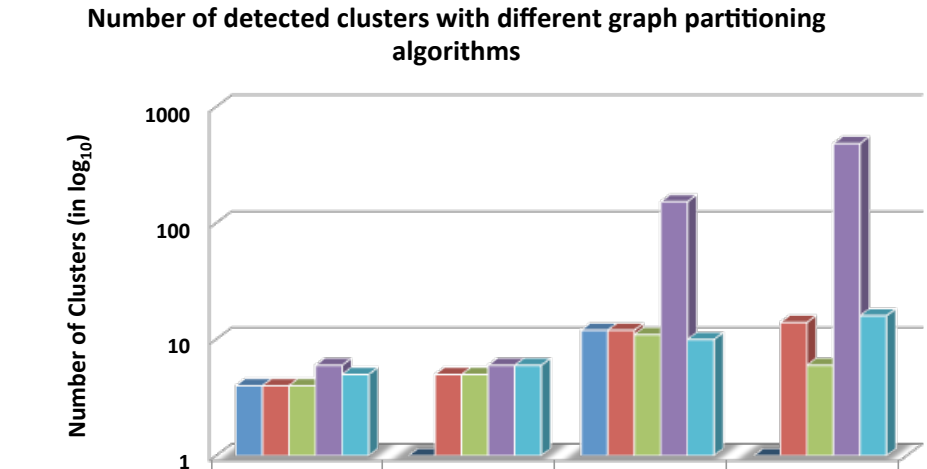
Leading Eigenvector [93] also uses modularity, expressed as Eigenvalues and Eigenvectors of a matrix. This matrix, which is independent of any network partitioning, can be treated as a network characteristic. Leading Eigenvector uses this matrix in identifying communities as well as detecting bipartite or k-partite structure in networks. Infomap [115] uses a measure called description length to find communities. Description length is defined as the expected number of bits per vertex required to encode the path of a random walk. Infomap depends on finding the shortest description length for a random walk on the graph to find communities.

LinLog layout [95, 96] uses an energy model to create a force-directed graph layout, a layout which is shown to subsume Newman and Girvan's modularity measure. This algorithm tries to position nodes such that densely connected nodes are grouped together while weakly connected nodes are placed at distant locations.

Figure 4.7 shows the number of detected groups using each algorithm. The X-axis contains the four datasets and the Y-axis lists the number of detected clusters for each dataset using different clustering algorithms. The Infomap algorithm detected large numbers of clusters in the Social network dataset compared to other algorithms. The Y-axis uses a logarithmic scale to show the values and their differences. As seen in this figure, Spinglass detects four clusters in the AlgoViz DSN, zero clusters in the Ensemble DSN, 12 clusters in US airport network, and zero clusters in the

[5] `http://toreopsahl.com/datasets/#usairports`

[6] `http://toreopsahl.com/datasets/#online_social_network`



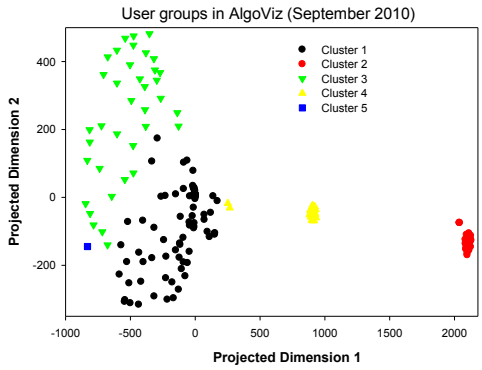
	AlgoViz - Sep10	Ensemble - Mar12	US airport	Social network
Spinglass	4	0	12	0
Multi-level	4	5	12	14
Leading Eignvector	4	5	11	6
Infomap	6	6	153	485
LinLog Layout	5	6	10	16

Figure 4.7: Community detection using graph partitioning algorithms.

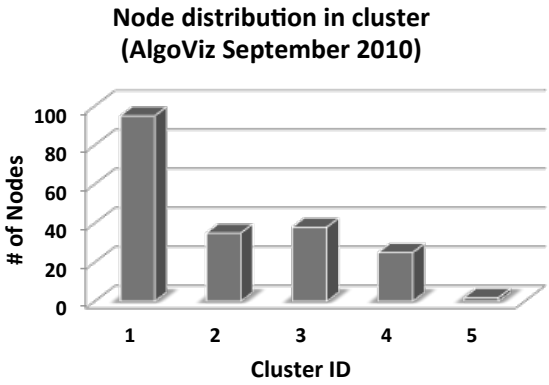
Social network dataset. Multilevel detects similar numbers of clusters, except where Spinglass found zero clusters in the Social network dataset, it is able to detect 14 clusters. Leading Eigenvector has similar results, detecting six clusters in the Social network dataset whereas Multi-level detected 14. The Infomap algorithm, on the other hand, detects a large number of groups compared to other algorithms for certain datasets (e.g., US airport, Social network). Multilevel, leading Eigenvector, and LinLog layout detect somewhat similar numbers of groups for all the datasets. Note that these three algorithms depend on some form of modularity. Of all the algorithms, only Spinglass sometime fails to detect groups within a network (e.g., Ensemble-Mar12, Social network).

For further analyses we choose to use the groups found by the LinLog layout algorithm. One of the reasons we select this algorithm is it uses modularity, which has been successfully used in other domains to detect communities. Although both the multi-level and Leading Eigenvector algorithms depend on some form of modularity, multi-level uses a hierarchical grouping approach which does not seem relevant for AlgoViz. The three main collections hosted in AlgoViz (i.e., catalog, bibliography, field reports) are not hierarchic in nature. On the other hand, because Leading Eigenvector needs to compute an Eigenvalue, it might not perform well for certain graphs. LinLog layout presents a suitable choice among the modularity-based algorithms we considered.

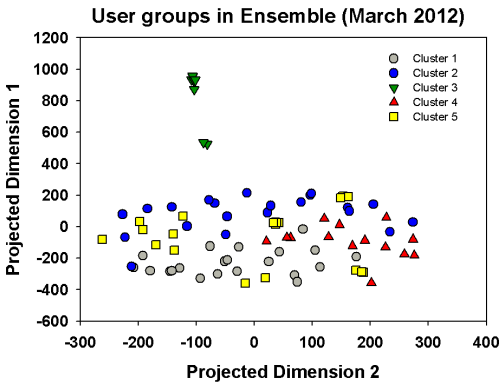
Using the results obtained from the LinLog layout algorithm on the AlgoViz September 2010 dataset, we project the data points of the clusters into a two-dimensional space. Figure 4.8 (i) shows the projected points in a 2D space. Each point represents a user in a cluster and each cluster is presented with a specific color. The user (i.e., node) distribution in the clusters is given beside the cluster



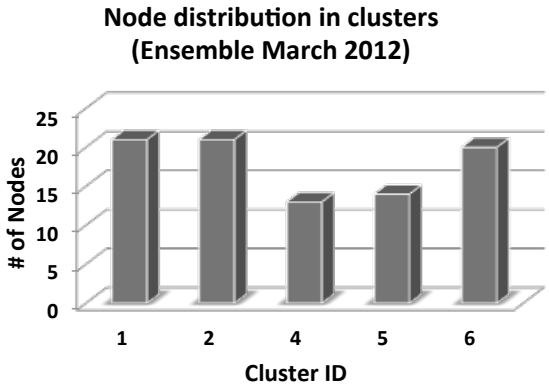
(i) Clusters in AlgoViz DSN (September 2010)



(ii) Nodes per group (September 2010)



(iii) Clusters in Ensemble DSN (March 2012)



(iv) Nodes per group (March 2012)

Figure 4.8: Groups of users in the AlgoViz and Ensemble DSNs using LinLog Layout.

plot (see Figure 4.8 (ii)). The figure shows that most of the clusters consist of more than 20 users. Also, there are clusters with low numbers of users, indicating possible outliers (e.g., Cluster 5 in AlgoViz September 2010 has 2 users). While most of the clusters of the September 2010 DSN are closely situated (see Figure 4.8 (i)), Cluster 4 is placed at a larger distance than the rest of the clusters. A possible reason of this could be that this cluster is made of users who saw different types of resources than most other users. Further analyses on the topic of this cluster in the next section points in this direction.

Figure 4.8 (ii) shows that the first cluster has close to one hundred users. In Figure 4.8 (i) the first cluster is represented using black circles. The second cluster is denoted by red circles and has more than 30 users (see Figure 4.8 (ii)). Clusters 3 and 4, represented by green and yellow triangles respectively (see Figure 4.8 (i)), have more than 20 users in them (see Figure 4.8 (ii)). The last cluster, presented by blue squares, has less than ten users.

Figure 4.8 (iii) and (iv) shows similar plots for the Ensemble March 2012 dataset. In Ensemble we see a more uniform distribution of users in the clusters. Clusters 1, 2, and 6 have at least 20

users whereas Clusters 4 and 5 have more than ten users (see Figure 4.8 (iv)). Most of the clusters also are placed closed to each other in Figure 4.8 (iii) except for Cluster 3. This indicates that the partitioning between the clusters may not be rigid and only Cluster 3 may show clear separation with the other clusters.

Building the DSN and partitioning the network gives us a glimpse of user groups present in these DLs. While one group might be dominating in one edu-DL (e.g., Cluster 1 in AlgoViz) another edu-DL may contain equally weighted groups (e.g., Ensemble). Even for a certain edu-DL, the groups and node distribution would change from time to time following user trends.

4.3.4 Topic modeling – finding community interests

Partitioning the DSN alone does not provide us with helpful information. The groups found by partitioning the DSN can be analyzed from different perspectives. Such analyses will help us to detect the characteristics of the groups. As an example we show how topic detection can be used to identify user interests in those groups using the AlgoViz September 2010 and Ensemble March 2012 datasets. Each cluster or group corresponds to a set of users who viewed some pages. For each group, we apply a topic modeling algorithm on the page titles viewed by the users of that group. Table 4.3 provides a overview of modeled topics for AlgoViz September 2010 DSN by two commonly used topic modeling algorithms, LSI [37] and LDA [14].

The first column of Table 4.3 lists the modeling algorithms: LDA and LSI. The second column shows the cluster IDs for the groups detected in the AlgoViz September 2010 DSN. For each cluster in the second column the topic with most coverage is listed in the third column. For each clustering algorithm we modeled five topics using both LDA and LSI since there are five clusters in the DSN. The reason for selecting the same number of topics is that it would allow us to see if each cluster is dominated by one specific topic. Of the five modeled topics we selected the topic with most coverage for any given cluster. The third column shows the ID of this topic while the fourth column shows the coverage of the topic in that cluster. For example, from row one we see that when using LDA,

Table 4.3: Significant topics for each group with LDA and LSI (AlgoViz Sep-10 DSN).

	Cluster ID	Topic ID	Topic coverage(%)	Words appearing in the topic
LDA	1	4	85.4	sigcse, algoviz, awards, publication, softvis, winners, call, acm
	2	5	89.79	biblio , tagged, export, catalog, entries, av, popup, trees
	3	3	86.67	biblio, forum, export, popup, author
	4	3	82.15	biblio, forum, export, popup, author
	5	1	91.66	biblio, export, author, popup, tagged, bibtex, catalog, entries
LSI	1	3	58.73	forum, sigcse, algoviz, publication, venues, softvis
	2	1	90.21	biblio, export, tagged, popup, author, catalog, entries, bibtex
	3	1	95.2	biblio, export, tagged, popup, author, catalog, entries, bibtex
	4	3	80.72	forum, sigcse, algoviz, publication, venues, softvis
	5	1	97.21	biblio, export, tagged, popup, author, catalog, entries, bibtex

Table 4.4: Topic distribution using LDA for AlgoViz September 2010 DSN, T=5 Topics.

Cluster ID	Top Topic (1)	Distribution	Top Topic (2)	Distribution	Top Topic (3)	Distribution
1	4	85.43%	5	3.64%	1	3.64%
2	5	89.79%	3	2.57%	1	2.56%
3	3	86.67%	1	3.37%	5	3.35%
4	3	82.15%	5	4.46%	1	4.46%
5	1	91.66%	5	2.0%	3	2.0%

for Cluster 1, Topic 4 covers 85.4% of the words appearing in that cluster. Similarly, Topic 5 has a higher topic coverage (89.79%) in Cluster 2. Among the four topics detected by LDA in the five clusters, Topic 4 links to words related to awards given to AVs. Topics 1, 3, and 5 all have common terms such as *biblio, export* that link more to the bibliography section and the activities related to a bibliography. However, Topic 1 also contains word such as *catalog* and *entries* that relate to the AV collections in AlgoViz. Moreover, Topic 5 also contains the word *trees* - likely referring to the AVs related to trees.

Likewise, using LSI, for Cluster 1, Topic 3 covers 58.73% of the words appearing in that cluster. Note that the clusters are the same for both LSI and LDA. The last column in this table shows some of the words appearing in the dominating topic (i.e., column three). Some of the words appearing in Topic 4 for LDA are *SIGCSE, algoviz, awards*, and *publication*, which relate to awards given to catalog entries at various venues (e.g., SIGCSE, SoftVis). The first topic detected by LSI contains the words *biblio, export, tagged*, and *popup*, which is likely to be related to events where users export bibliography entries tagged with particular author names. Topic 1 of LSI also contains words related to AV catalog entries. Unlike LDA which detected four topics in the five clusters, LSI detected two topics that cover the five clusters - Topics 1 and 3. Topic 1, described earlier, contains a mix between bibliography and catalog entry sections within AlgoViz. Topic 3 contains words *forum, SIGCSE, algoviz*, and *publication*, which are more general and not tied to any specific collection or resources.

Overall, in the five clusters of AlgoViz September 2010 DSN, LDA detected four topics that have more than 50% coverage in the clusters (i.e., Topics 1, 3, 4, and 5) whereas LSI detected two such topics (i.e., Topics 1 and 3) (see column three in Table 4.3). In a perfectly partitioned network each cluster would contain users who viewed resources that are unique to that cluster only. When topics are modeled in such clusters the dominant topic for each cluster also would be unique. From Table 4.3, we see that the topics of LDA are more in sync with the partitioned networks for this particular network and the associated clustering approach.

To further analyze the topics discovered by the LDA approach, we examined the top three topics appearing in each cluster. Table 4.4 shows the distribution of the top three topics for each of those five clusters using LDA. For example, in Cluster 4, Topic 3 covers 82.15% of the content (e.g., title of the pages visited by the cluster members). For this cluster the two other top topics are Topic 5 (4.46%, see column five in Table 4.4) and Topic 1 (4.46%, see column seven in Table 4.4). We see that for each cluster the top topic always covers more than 80% of the cluster content. We also see that one topic may be dominant in multiple clusters. For example, Topic 3 is dominant in both Clusters 3 and 4.

Table 4.5: Topic distribution using LDA for Ensemble March 2012 DSN, T=5 Topics.

Cluster ID	Top Topic (1)	Distribution	Top Topic (2)	Distribution	Top Topic (3)	Distribution
1	2	43.20%	5	27.10%	–	–
2	2	38.60%	5	28.10%	3	22.30%
3	1	49.40%	–	–	–	–
4	2	41.30%	5	28.40%	–	–
5	2	41.60%	5	21.80%	–	–

Results of similar analyses on the Ensemble DSN are described in Table 4.5. For each cluster (in the first column), we list the top three topic IDs along with the percentage of text the topic covers. The empty cells indicate that none of the discovered topics match the rest of the texts in the cluster. For example, with Cluster 3, we see that Topic 1 covers 49% of the text appearing in that cluster. However, none of the other topics were able to cover the remaining 51% of the text appearing in that cluster.

As we see in Table 4.5, Topics 1 and 2 are the prominent topics for the five clusters found in the March 2012 Ensemble DSN. Clusters 1, 2, 4, and 5 all have Topic 2 as the most dominating topic while Cluster 3 is dominated by Topic 1. Although we are referring to the first topic as the most dominating topic for a given cluster, none of these dominating topics cover more than 50% of the cluster content. Also, although we found five partitions in the DSN, from the topics identified we are able to detect two groups of users: the group who viewed contents related to Topic 1 and the group who viewed content related to Topic 2. Table 4.6 lists some of the words appearing in the top three topics in the Ensemble DSN. Words from Topic 1 point to one of the groups in Ensemble called *Passion, Beauty, Joy and Awe* and activities related to groups in general, such as *join* and *subscribe*. Topic 2 points to pages with terms *design, history, institute*, and *cs*, which likely refers to the history of computer science. Topic 3 includes words such as *computer, science, software*, and *java* which links more to areas in computer science.

The use of topic modeling has two benefits. On the one hand it acts as an analytic tool, providing us with the information on the characteristics of the groups and their interests. At the same time this can be used to identify if the partitioning of the DSN was rigid. As we stated earlier, ideally, for a well partitioned network, each partition (i.e., cluster) should be dominated by a unique topic. If, however, two or more partitions (i.e., clusters) share the same topic as the most dominant topic, then the quality of the partitions may not be satisfactory. In that case, further analyses can be carried out to find better partitions. In our case, the use of topic modeling was more exploratory than necessary. In the later chapters where we use the findings of the DSN we only rely on the

Table 4.6: Top three topics in the Ensemble March 2012 DSN using LDA.

Topic ID	Some words appearing in the topic
1	computing, group, og, joy, join, beauty, cs, subscribe, ensemble
2	taxonomy, term, feed, subject, cs, design, image, history, institute, amp
3	taxonomy, site, community, computer, java, software, science, data

groups detected by the network partitioning.

4.4 Summary

In this chapter we describe how log data can be used to generate deduced social networks within educational DLs. We also show how these DSNs can be partitioned to detect groups of users. Further, we use topic modeling to identify the group characteristics. In the absence of active user communities these latent user groups can help us improve DL services by incorporating user trends into the services or by sharing usage patterns with users. Our approach is generic enough that it can be implemented in other types of DLs. The focus of this chapter is to define a DSN, show how DSNs can be modeled using data from real edu-DLs, and present exploratory analysis on those DSNs. The applicability of the findings are described in the subsequent chapters where we use the groups from the DSNs to improve ranking of search results and provide recommendations to users.

Chapter 5

Revised Ranking in an Educational DL

In Chapter 4 we described the process of generating a deduced social network and then partitioning the network. As a result of network partitioning, we found a set of groups. Findings from the network and group analyses can be used in various ways. This chapter shows how DSNs can be used to improve the content ranking system for an educational DL. In particular, we used DSNs to improve the search rankings for Algorithm Visualizations in the AlgoViz catalog.

5.1 AlgoViz ranking

AlgoViz[1] is an educational portal with an online community. AlgoViz has a comprehensive collection of metadata on algorithm visualizations (AVs) and a bibliography related to algorithm visualizations. Each of these collections has more than 500 entries. Each AV entry in the catalog contains metadata for the associated AV, including a link to the original AV. Other fields in the metadata include author, project it is part of, language of the AV, institution, date it was first published, date it was last modified, topic it addresses, and delivery method such as Java applet (see Figure 5.1). Along with the metadata many of the AVs also have detailed information including description, evaluation, usage notes, field reports, and bibliography entries linked to it. Each AV also contains a four-level recommendation that shows different areas where the AV might be used.

During browsing and searching of catalog entries in AlgoViz, users are presented with a list of AVs in a succinct form that shows limited information about the rating, recommendation, topic, and delivery method (see Figure 5.2). This list is sorted according to a customized weight computed for each catalog entry in AlgoViz. AlgoViz is implemented using the Drupal[2] content management system. The default sorting scheme for Drupal depends on generic fields of a resource such as recency of the resource, number of comments, title, etc. When used alone these generic fields fail to convey information on the state of an AlgoViz catalog entry — whether the entry is working, what the delivery method is, what format it is in, what it is good for. This information, which can be useful for the user, can be found in the custom attributes of each catalog entry. AlgoViz uses custom

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

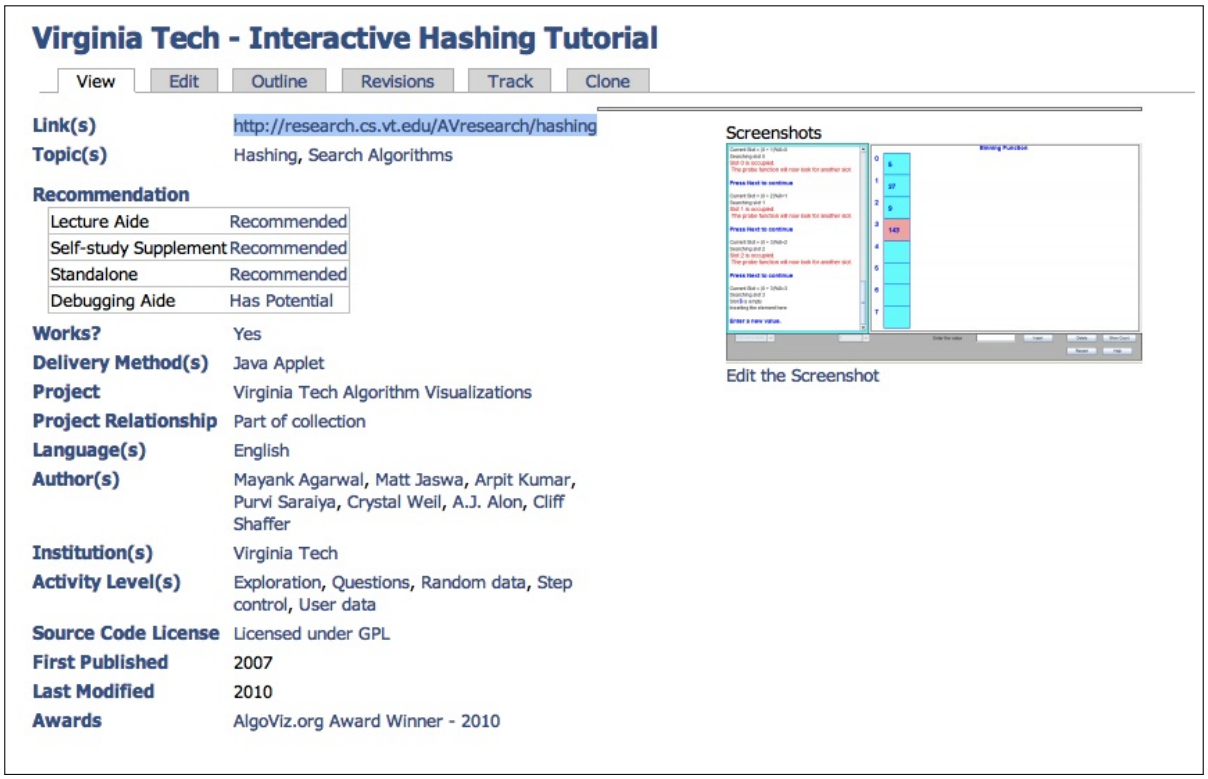


Figure 5.1: An AV in AlgoViz.

scoring as a way to incorporate these AlgoViz-specific custom attributes into the overall ranking of catalog entries. This is done by providing certain weights to the custom attributes. These weights lead to a custom score which is used to rank catalog entries. Each custom field is given a specific weight by domain experts. We will refer to this weight as *custom score* and the fields that are used to calculate the weight as *custom fields*.

Among the various attributes of an AV only seven are used to compute the *custom score*. The field *works* indicates if the AV works or not. A working AV receives 20 points towards its custom score. There are four levels of *recommendations* (i.e., *recommended, has potential, unrated*, and *not recommended*) for four different purposes for which the AV is evaluated by domain experts (i.e., lecture aid, self-study supplement, standalone, debugging aid). There are 20 points for the first *recommended* and six points for the subsequent *recommended*. Each *has potential* receives five points, *unrated* receives one point, and *not recommended* receives two negative points. AlgoViz lists a number of awards given to outstanding AVs by both AlgoViz and other educational institutes/conferences (e.g., Koli calling educational tool award). Any *award* of an AV receives 15 points. Five points are awarded for any *reference* or *field report* linked to the AV. Support for multiple *languages* in an AV receives five points. Based on the type of the *delivery method* of an AV it can receive one or two points. Lastly, the field *activity level* which indicates the type of the AV (e.g., animation, slide show) as well as the interactivity level (e.g., step control, questions) is given two to five points depending on the activity level. The custom ranking scheme is summarized in Equation 5.1.

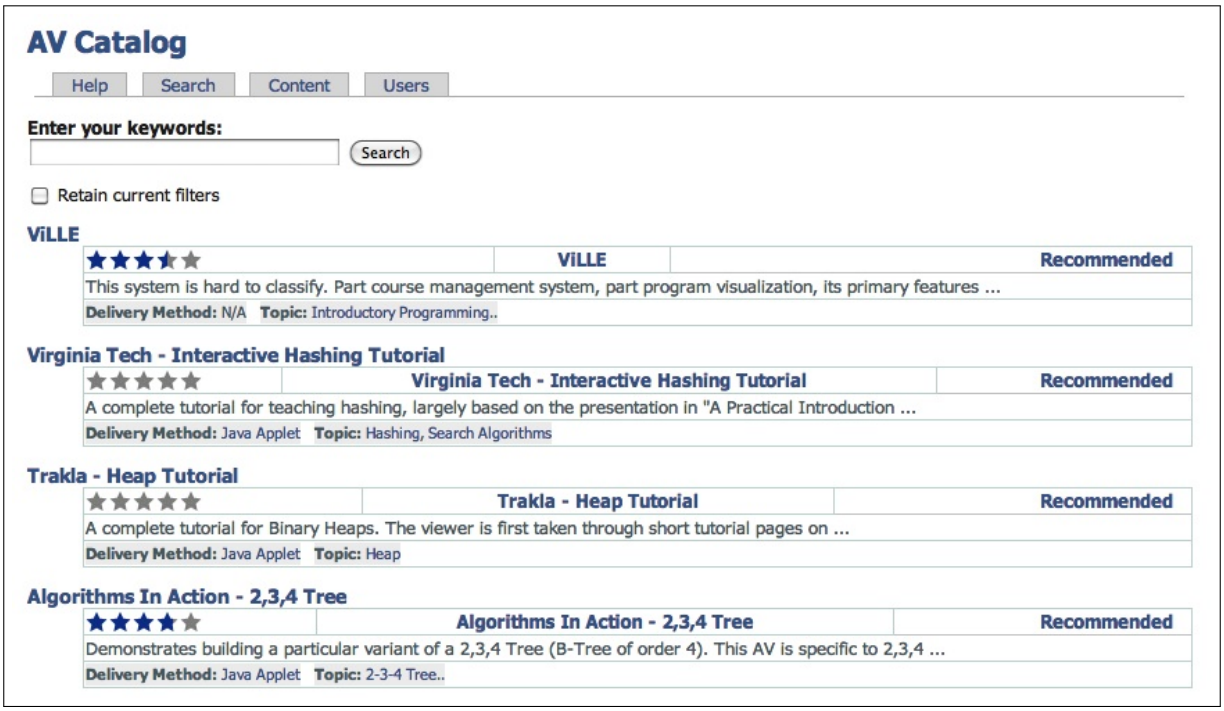


Figure 5.2: AlgoViz catalog.

$$\begin{aligned}\text{Custom score} = {} & \textit{Works} * 20 + \text{First } \textit{Recommended} * 20 + \text{Additional } \textit{Recommended} * 6 + \\ & \text{Each } \textit{Has Potential} * 5 + \text{Each } \textit{Unrated} * 1 + \text{Each } \textit{Not Recommended} * -2 + \\ & \text{Any awards} * 15 + \text{Any } \textit{Field Report} \text{ and/or Reference} * 5 + \text{Multiple } \textit{Language} * 5 + \\ & \text{Any delivery method with } \textit{Java} \text{ but not } \textit{Java-webstart} * 2 + \\ & \text{Delivery method with } \textit{Java-webstart} * 1 + \text{Activity level has } \textit{Predictions} \text{ or } \textit{Exploration} * 5 + \\ & \text{Activity level has } \textit{Questions} * 2 \end{aligned} \tag{5.1}$$

Two services within AlgoViz, browsing and searching, make use of the *custom score*. During AV catalog browsing the AV list is sorted based on this *custom score*. As for searching, AlgoViz provides two types of searches to its user. The default Drupal search is a site-wide search that returns all content types (e.g., AV, bibliography, forum post) matching the query terms. There is also an advanced searching option provided in the catalog page. This search builds upon the Solr framework[3]. The Solr-based search is performed only on the AV catalog entries and the results are sorted according to the *custom score* of the AVs. Since the ranking of Solr-based search is customized, we refer to this ranking as the *custom ranking*.

While the *custom ranking* depends solely on Equation 5.1, Drupal ranking uses keyword relevance,

[3] `http://lucene.apache.org/solr/`

Table 5.1: Drupal ranking factors and weights in AlgoViz.

Factor	Weight
Keyword relevance	8
Recently posted	5
Number of comments	4
Number of views	6

recency of the resource posted, number of comments, and pageviews[4]. Table 5.1 shows the current weight associated with the four fields used in the Drupal ranking within AlgoViz. These are default weights for the Drupal search framework. The weights control which and how much of certain properties of the content should be valued while ordering the results. Higher numbers indicate more influence in the ordering.

While Drupal allows site administrators to change the weights of the listed factors, it does not provide a way to include any custom fields in the ranking. The default Drupal search module in Drupal 6 (the version used by AlgoViz) also lacks a faceted searching option. Faceted search allows users to filter the search results according to the different attributes of an entry which act as facets. These factors lead us to use the Solr search framework for AlgoViz. While Solr provides faceted search we need to customize the Drupal module for Solr in order to include the custom fields of AVs as facets. We also customize the framework so that the ranking is done based on the *custom score*.

One potential drawback of the AlgoViz custom ranking is that it lacks user trends (e.g., pageviews). DSNs are built using log data, which contains user behavior. Various network analyses on the DSNs can reveal interesting user trends as seen in Chapter 4. We believe the DSN holds potential to improve the performance of *custom ranking* by including social trends into the ranking.

5.2 Revised ranking

Ranking is a crucial part of any query system. Figure 5.3 shows how ranking and searching are used for a query. Different searching frameworks depend on different measures to find the most relevant resources for a given query. The results are ranked before presenting them to the user. This chapter addresses the ranking section (appearing within the dotted red lines in Figure 5.3) of a query processing system. Most retrieval methods rank the results by placing the presumed most relevant content on the top of the list. Text-based ranking mostly depends on term frequency and text similarity. Link-based ranking on the other hand relies on the links within and across documents. The two existing searching systems in AlgoViz uses two different search frameworks. The default Drupal search results are ranked using the default weighting scheme as described in the previous section. The ranking of the Solr system, described in Table 5.1, is overridden by the AlgoViz custom score.

One of the main differences between these two approaches is that the *custom fields* related to the *custom score* are AlgoViz-specific fields which lack usage information, whereas Drupal includes

[4]Pageview: Loading of a webpage.

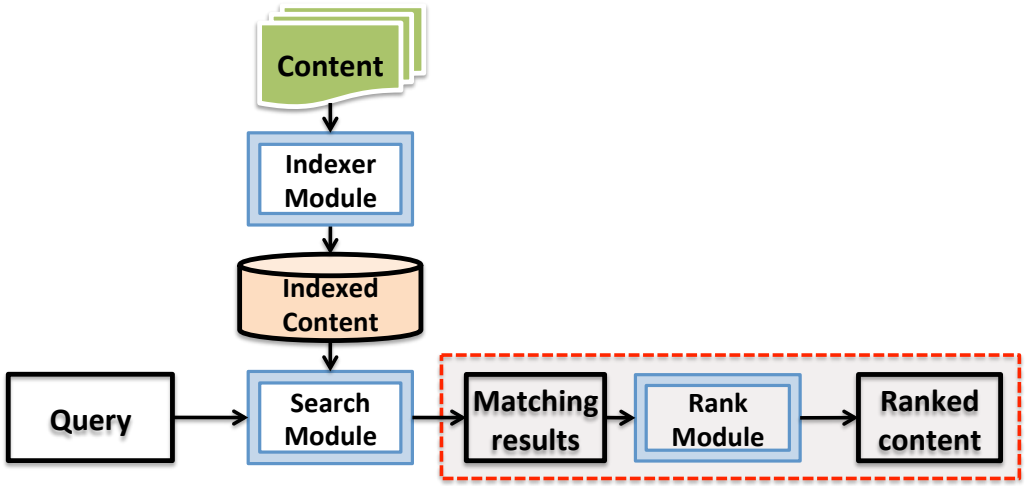


Figure 5.3: Ranking resources during search.

number of views (see Table 5.1). DSNs built with pageviews capture such information and can be useful in including usage information into the *custom ranking*.

Pageview counts is a common measure used to gauge user interest. However, pageview counts fail to show if a resource was viewed more by a certain set of users with similar interests than by all users in general. Partitioning the DSN gives us latent groups of users with different interests. Using these groups found in the DSN, we can identify if a resource is of interest to a specific group or to a wider audience (e.g., all groups). To capture this information on the variety of users of a resource, we introduce the *social interest coefficient*, s.

For each resource, s will reflect if the resource (e.g., catalog entry, bibliography) was viewed by different types of users. More weight will be given to a resource that appears in more groups — indicating it is of interest to a wider set of users. This coefficient reflects the fraction of groups whose users viewed the resource. The social interest coefficient is defined as:

$$s = \frac{\#\text{ of groups containing the resource}}{\#\text{ of total groups}} \tag{5.2}$$

For example, the AlgoViz DSN of September 2010 has five groups (i.e., clusters) of users. For each group, we have a list of resources seen by the members of that group. If a resource re_i was seen by the members of all five groups then s_{re_i} for that resource is calculated as $s_{re_i} = \frac{5}{5} = 1$

Similarly, if a resource re_j was seen by the members of only one group then s_{re_j} for that resource is calculated as $s_{re_j} = \frac{1}{5} = 0.2$

The value of s will range from 0 to 1 where 1 indicates that the resource is viewed by all the groups and a value of 0 will indicate that no member in any group viewed the resource.

For any given resource, the *social interest coefficient* shows the fraction of groups whose users viewed

that resource. It however fails to assign any weight to this fraction. A resource that is viewed one time in all groups will have the same s as the resource that is viewed more than once in each group. To capture this information we propose a *weighted interest coefficient*, p. For each resource viewed by the users in a group, p will indicate if the resource was viewed by more users of that group. For a given group g_j, for any given resource re_i we define p as:

$$p = \frac{\text{\# of users who viewed } re_i \text{ in group } g_j}{\text{\# of users who viewed the most-viewed resource(s) in group } g_j} \tag{5.3}$$

As an example, assume there are five resources viewed by the members of group 1. Along with each resource, assume we also know how many users viewed each resource:

$re_1 = 5$,

$re_2 = 1$,

$re_3 = 2$,

$re_4 = 10$,

$re_5 = 3$

Thus, re_4 is the most viewed resource (i.e., 10 users viewed re_4). Then p for each resource of group 1 will be:

$p_{re_1} = \frac{5}{10} = 0.5$

$p_{re_2} = \frac{1}{10} = 0.1$

$p_{re_3} = \frac{2}{10} = 0.2$

$p_{re_4} = \frac{10}{10} = 1$

$p_{re_5} = \frac{3}{10} = 0.3$

Thus, in a certain group, resources that are viewed by more users will be weighted more compared to less viewed resources.

When used together, the two coefficients s and p provide information on which groups of users viewed a resource and how the pageview of that resource contributes to the overall pageview of all the resources in any given group. The behavior of isolated users who are not part of any group do not reflect a common navigation trend. Hence information of isolated users is not included on either of these two coefficients. With the two DSN-derived coefficients, we propose a DSN-based ranking schema defined as:

$$\text{DSN score}_{re_i} = s_{re_i} \times \sum \{p_{re_1}, p_{re_2}, \ldots, p_{re_n}\} \tag{5.4}$$

where n is the number of groups in the DSN.

While custom ranking uses custom scores of catalog entries (see Equation 5.1), DSN-based ranking depends solely on DSN-derived information. Both these rankings use two different types of information — custom ranking contains expert's opinion on the equality and usefulness of the catalog entry and DSN-based ranking shows level of users' interest in that resource. When used together these two sets of information can provide us with better ranking compared to the two separate rankings (i.e., custom ranking and DSN-based ranking).

Thus, we propose a ranking that builds upon the custom ranking and adds usage information with the help DSN-based ranking. Revised ranks are computed based on the revised score. For any given resource re_i we define the revised score as:

$$\begin{aligned} \text{Revised score}_{re_i} &= \text{Custom score}_{re_i} + \text{DSN-score}_{re_i} \times 100 \\ &= \text{Custom score}_{re_i} + s_{re_i} \times \sum \{p_{re_1}, p_{re_2}, \ldots, p_{re_n}\} \times 100 \end{aligned} \tag{5.5}$$

where n is the number of groups in the DSN.

In most cases, the values of s and p are fractions, resulting in a fractional DSN-score. Custom scores however, produces round numbers usually ranging from around 7 to 90. In the revised ranking, in order to reflect the small changes in the DSN-score next to the custom scores, we multiply the DSN-score with 100.

We hypothesize that *revised ranking* will perform better than *custom ranking* in identifying user interests. In the next section we analyze a series of cases to test the hypothesis. We create a benchmark ranking and compare the performance of *revised ranking* with the performance of *custom ranking* and Drupal ranking in AlgoViz.

5.3 Rank evaluation

Our hypothesis is that revised ranking has the potential to better predict user interest compared to the other rankings in AlgoViz. To test this, we compared the performance of DSN-based ranking and revised ranking which uses the DSN-derived coefficients along with the custom score, against the performance of the two other ranking systems in AlgoViz: Drupal and custom ranking. One of the major differences between these two rankings is that Drupal uses pageviews but custom ranking does not. The DSN-derived coefficients are related to pageview. To see how pageview alone performs in ranking resources, we generate a solely pageview-based ranking and test it with the benchmark. However, it seems likely that solely pageview-based ranking is not preferable for a number of reasons. Pageviews can be generated through various activities originating within or outside the DL. For example, an email or a webpage with a link to a particular resource within an end-DL would generate pageviews for that resource. Thus, pageviews can produce inaccurately biased ranking. Also, the act of viewing a page related to an educational resource does not provide any feedback on the potential usefulness of a resource. Contrary to pageview, in this scenario, a click on the URL of the resource better portrays user interest in that resource. The pageviews for each catalog entry were computed from the log for the duration of the DSN (i.e., Spring 2011).

As a measure for user interest we introduce the *log-based rank* that depends on the *AV access* field. AV access is the number of clicks on the outgoing URL of a catalog entry in AlgoViz (from June 2011 to September 2013). The reason we select this field to generate the benchmark ranking is that while pageview shows initial interest in an AV, a click on an outgoing link shows further interest and can be seen as positive feedback on the potential usefulness of an AV. It is the best objective measure available that shows a catalog entry is really what the user is searching for.

We follow a case study approach for our evaluation. We randomly selected ten query terms from AlgoViz and compared the ranking of resultant AVs under the three different ranking systems. While we do propose a definition for revised ranking that incorporates DSN-derived parameters in the existing ranking, better models and approaches may exist. However, our goal for this chapter is to show the potential of DSN in improving an existing DL service (i.e., ranking) in a particular edu-DL: AlgoViz.

5.3.1 Methodology

In this section we briefly describe the methodology for data collection and analysis. We begin by randomly selecting ten queries from the AlgoViz query log. While selecting these queries, one criteria was that no two queries should be overlapping. Thus for query terms *AVL* and *AVL tree*, we only selected one (e.g., AVL) for further analysis. Also, we discarded the query terms that resulted in AVs with no pageviews in the DSN we used (AlgoViz Spring 2011).

Once we selected ten unique queries, for each query we perform two searches in AlgoViz, one using the default Drupal search and the other using the Solr search. We filtered the first search performed by Drupal so that only catalog entries were listed. This set of results was sorted based on the Drupal sorting schema described in Table 5.1. A similar search on the AV catalog was performed using Solr search. Solr search results were sorted according to the *custom score*.

The number of results varied depending on the search system. For example, for the query *Binary search*, the generic Drupal search returned 28 catalog entries while the Solr search returned 24 catalog entries. The Solr search results were sorted based on the custom score (described in Section 5.2). For each catalog entry returned by the Drupal search, we gather information on its *AV access*, *custom score*, s, p, and pageview. AV access contains the number of clicks on the outgoing URL of each AV from June 2011 to September 2013. The current system that tracks clicks on the outgoing link provides a cumulative value. It does not capture the time of each click. Hence anytime we process the clicks on the outgoing links we must take the total number of clicks on any outgoing link. The social interest coefficient (s) and the weighted interest coefficient (p) are calculated based on the Spring 2011 DSN of AlgoViz. Pageview also was computed from the same time frame using cleaned log data. For each query term we compute the revised ranking based on Equation 5.5. The computation of the coefficients and the pageviews from the DSN are not computationally intensive for the particular DSN we use (AlgoViz Spring 2011) which is one of the largest among the four DSNs we examined. However, finding the rank of the pages of a search is a manual and time consuming process that includes performing the search and manually processing the results (e.g., rank, page ID).

Various rank correlation measures are used to test the relationship between two ranks. Of those we

use Spearman's rho, Kendall's tau, and Pearson's rank correlation coefficient to test the performance of different rankings. We use log-based rank (see Section 5.3) as the benchmark and compare five other rankings: Drupal rank, *custom rank*, *pageview-based rank*, *revised rank*, and *DSN-based rank*, with the log-based rank. Details of the evaluation methods will be discussed next.

5.3.2 Spearman's Rho

Spearman's rank correlation coefficient, also known as Spearman's rho [127], is the rank correlation coefficient between n pairs of items. Spearman's rho is calculated as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{5.6}$$

where d is the difference between the matched ranks. The value of Spearman's rho ranges from -1 to 1. A value of 0 indicates no correlation. A value from 0.5 to 1 indicates strong positive correlation, 0 to 0.5 weak positive correlation, 0 to -0.5 weak negative correlation, and -0.5 to -1 strong negative correlation.

5.3.3 Kendall's Tau

Another rank ordering evaluation method that is often used to test association between variables is called Kendall's tau [66]. It is defined as:

$$\tau = \frac{(\text{number of concordant pairs}) - (\text{number of discordant pairs})}{\frac{1}{2}n(n-1)} \tag{5.7}$$

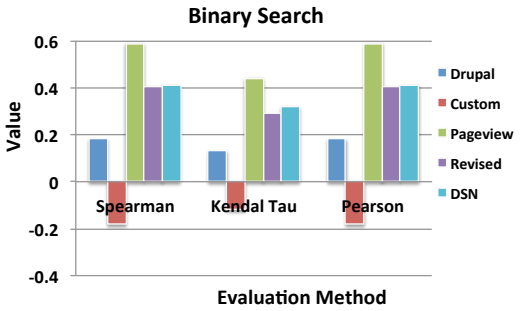
where n is the number of items.

A pair is said to be concordant if $X_i < X_j$ implies that $Y_i < Y_j$, for $i \neq j$. The pairs are said to be disconcordant if $X_i < X_j$ implies that $Y_i > Y_j$, for $i \neq j$. The value of τ ranges from -1 to 1. In the case where there is no correlation the value of τ is 0.

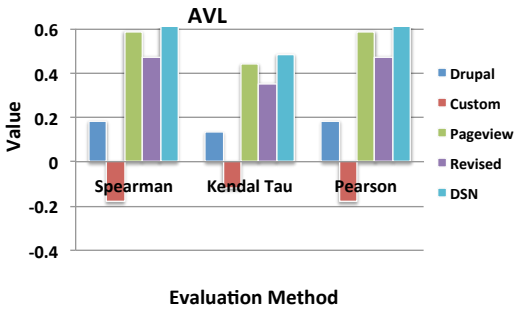
5.3.4 Pearson's Correlation Coefficient

Pearson's correlation measures the strength of linear relationship between two sets of data X_i and Y_i [127]. It is defined as:

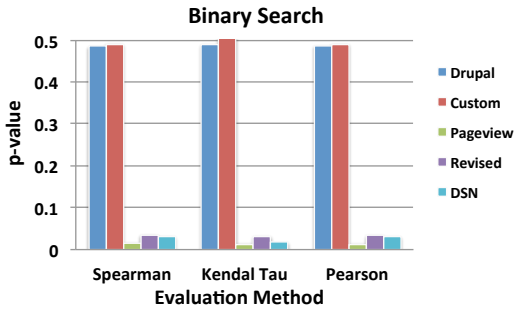
$$r = \frac{1}{n-1} \sum_{i=1}^{n} \frac{(X_i - \bar{X})(Y_i - \bar{Y})}{S_x S_y} \tag{5.8}$$



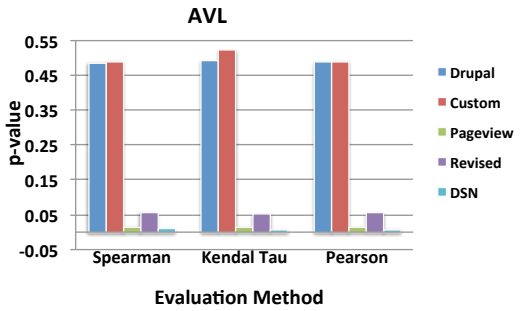
(a) Rank correlation coefficients



(c) Rank correlation coefficients



(b) p-values of rank correlation coefficients



(d) p-values of rank correlation coefficients

Figure 5.4: Rank correlation for query terms *Binary Search* and *AVL*.

where $\bar{X}$ is the mean of X, S_x denotes the standard deviation of X, and n is the number of items.

The value of r ranges from -1 to +1. A value of zero indicate no linear correlation whereas a value of 1 indicates perfect positive correlation. A value of -1 indicates a perfect negative correlation.

5.3.5 Results

Figure 5.4 shows the performance of the five ranks compared to the benchmark rank using query terms *binary search* and *AVL*. The five ranks are the Drupal rank, custom rank based on AlgoViz-specific fields, pageview-based rank, our proposed revised rank, and lastly the ranking done solely on the AlgoViz Spring 2011 DSN following Equation 5.4.

Figure 5.4(a) shows a bar chart for each ranking system under a specific evaluation method for the query term *binary search*. The first group of bars represents Spearman's rho (ρ) under which the Drupal rank returns a value of 0.18 when compared to the benchmark *log-based rank*. *Custom rank* returns a ρ value of -0.18. The ρ value of pageview-based rank is 0.58, revised rank is 0.404, and DSN-based rank is 0.408. According to this chart, using Spearman's rho, pageview-based rank, revised rank, and DSN-based rank perform better than the other two ranks.

The next group of bars represent the results of Kendall's tau (τ) evaluation. According to this evaluation the τ value for Drupal rank is 0.13, custom rank is -0.11, pageview-based rank is 0.44, revised rank is 0.29, and DSN-based rank is 0.31. Thus Kendall's tau show similar results as the Spearman's rho. The last group of bars denotes Pearson's coefficients, the results of which are

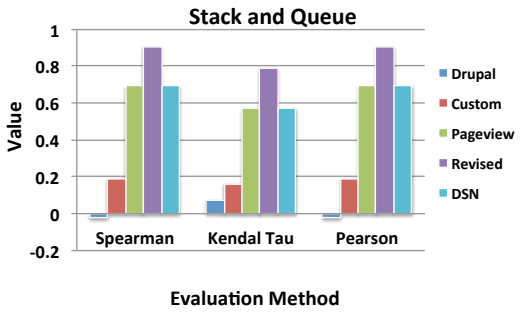
similar to Spearman's rho.

The coefficients alone are not useful in detecting the significance of the correlation. Figure 5.4(b) shows the corresponding p-values for each coefficient. A p-value of 0.05 or lower indicates significant correlation with the benchmark rank. This chart shows that using Spearman's rho, the p-values of Drupal and custom ranks are 0.48, hence failing to indicate any significant correlation between any of these ranks and the benchmark rank. The p-value of pageview-based rank is 0.01, revised rank is 0.033, and DSN-based rank is 0.031. Thus these three ranks show significant correlation with the benchmark rank.

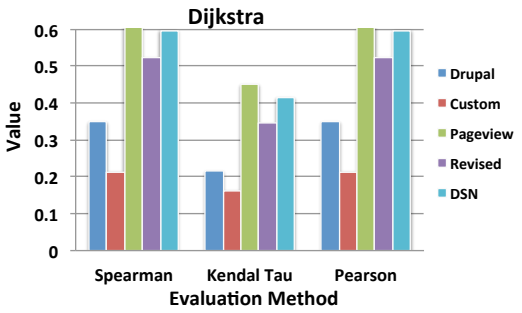
The second set of bars in 5.4(b) shows the p-values of the rank correlation coefficients using Kendall's tau. All the ranks except the custom rank have values similar to Spearman's rho. The p-value of custom rank using Kendall's tau is 0.5 which is slightly higher than the corresponding Spearman's rho (0.488). The third set of bars in 5.4(b) shows p-values for Pearson's correlation coefficient. The p-values are similar to those for Spearman's rho.

Figure 5.4(c) and 5.4(d) show the results using the query term *AVL*. The rank correlation coefficients using different evaluation methods are shown in Figure 5.4(c). The results of *AVL* are similar to *binary search*. Thus the coefficients (i.e., Spearman's rho, Kendall's tau, Pearson's coefficient) for Drupal rank and custom rank are smaller compared to the other three ranks: pageview-based rank, revised rank, and DSN-based rank. The corresponding p-values in Figure 5.4(d) show that according to all the evaluation methods, the last three ranks show significant correlation with log-based rank.

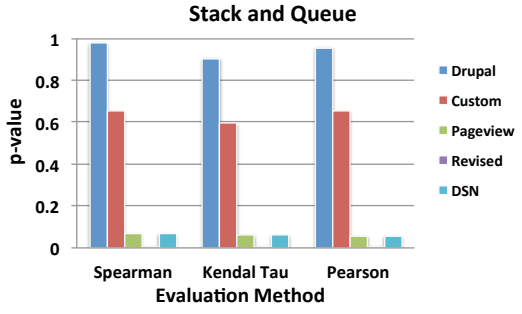
Out of the ten query terms, two other queries, *stack and queue* and *Dijkstra*, show similar trends



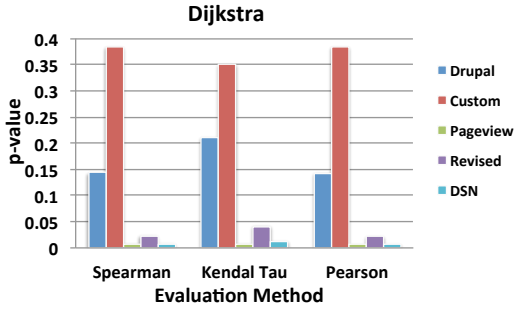
(a) Rank correlation coefficients



(c) Rank correlation coefficients

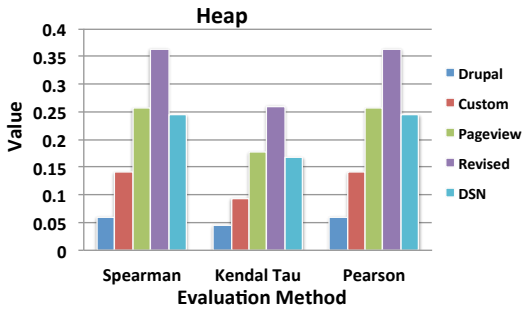


(b) p-values of rank correlation coefficients

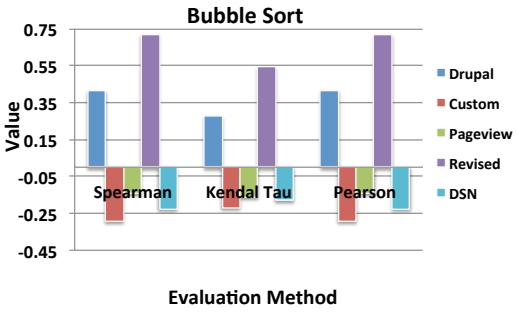


(d) p-values of rank correlation coefficients

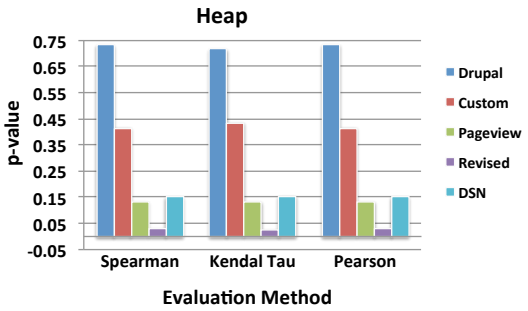
Figure 5.5: Rank correlation for query terms *Stack and Queue* and *Dijkstra*.



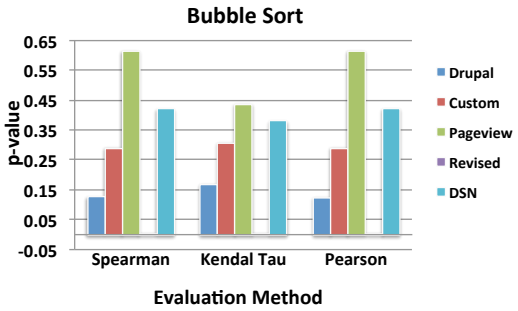
(a) Rank correlation coefficients



(c) Rank correlation coefficients



(b) p-values of rank correlation coefficients



(d) p-values of rank correlation coefficients

Figure 5.6: Rank correlation for query terms *Heap* and *Bubble Sort*.

where revised rank, pageview-based rank, and DSN-based rank show better performance compared to the other ranks. Figure 5.5(a) shows the rank correlation coefficients for *stack and queue*. According to Speaerman's rho, only the Drupal rank yields a negative coefficient (-0.02). The ρ value for custom rank is 0.19, pageview-based rank is 0.69, revised rank is 0.9, and DSN-based rank is 0.69. The τ value of Drupal rank is 0.07, custom rank is 0.16, pageview-based rank is 0.57, revised rank is 0.78, and DSN-based rank is 0.57.

The p-values for each coefficient of Figure 5.5(a) is shown in Figure 5.5(b). According to this figure, under all the evaluation methods, revised rank, pageview-based rank, and DSN-based rank show significant correlation with log-based rank.

The rank correlation coefficients for the query term *Dijkstra* is given in Figure 5.5(c). In this case, *custom rank* has the lowest coefficient in all the evaluation methods whereas the pageview-based rank and DSN-based rank have the highest coefficients. The corresponding p-values are given in Figure 5.5(d). It shows that revised rank, pageview-based rank, and DSN-based rank have p-values lower than 0.05, thus indicating significant correlation with log-based rank.

So far the four query terms we used showed similar trends, where revised rank, pageview-based rank, and DSN-based rank showed significant correlation with the benchmark log-based rank. In each of the four cases both Drupal rank and custom rank fail to show any significant correlation with log-based rank. We observed another trend where only revised rank showed significant correlation with log-based rank. Figure 5.6 shows this trend.

Figure 5.6(a) shows the rank correlation coefficients for the query term *heap*. Using Spearman's rho,

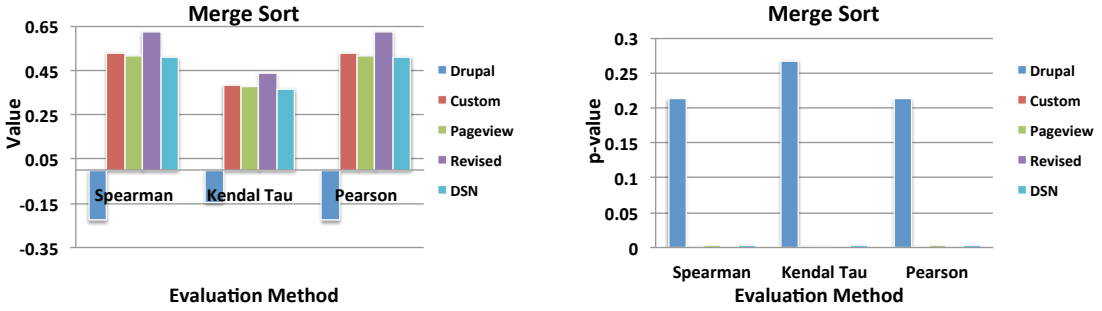
we see that the ρ value of Drupal rank is 0.05, custom rank is 0.14, pageview-based rank is 0.25, revised rank is 0.36, and DSN-based rank is 0.24. Out of the five ranks the highest ρ is achieved by the revised rank. A similar trend is seen using Kendall's tau, though the coefficients are different. The τ value for Drupal rank is 0.04, custom rank is 0.09, pageview-based rank is 0.17, revised rank is 0.26, and DSN-based rank is 0.16. The third set of bars show Pearson coefficients which are identical to Spearman's rho.

The associated p-values for Figure 5.6(a) are shown in Figure 5.6(b). Except for revised rank, all three ranks have p-values greater than 0.05 under all evaluation methods. The results indicate that when we use query term *heap*, only the revised rank is able to provide a ranking that is correlated with the benchmark rank.

The results of the query term *bubble sort* are shown in Figure 5.6(c) and Figure 5.6(d). Figure 5.6(c) shows the rank correlation coefficients. Using Spearman's rho Drupal rank achieves a ρ value of 0.41 whereas the ρ value of custom rank is -0.29, pageview-based rank is -0.14, and DSN-based rank is -0.22. The highest ρ value is achieved by revised rank which is 0.7. Both Kendall's tau and Pearson's coefficient show similar trends. The p-values in Figure 5.6(d) show that only revised rank has p-value less than 0.05 (i.e., 0.004) under all of the evaluation methods. Thus, when using query term *bubble sort*, only the revised rank shows significant correlation with log-based rank.

Next we present the results for *merge sort*. In Figure 5.7(a), for *merge sort*, the rank correlation coefficient for Drupal rank is -0.22 using Spearman's rho. The ρ value for custom rank is 0.52, pageview-based rank is 0.51, revised rank is 0.62, and DSN-based rank is 0.508. Similar to Spearman's rho, revised rank generates the largest τ value (i.e., 0.43) for the Kendall's tau method. The third evaluation, Pearson's coefficient, produces results similar to Spearman's tau. The p-values are given in Figure 5.7(b) which shows only Drupal rank fails to reflect any significant correlation with log-based rank. All of the other four ranks have p-values less than 0.05.

Drupal rank, pageview-based rank, and DSN-based rank perform poorly with the next query term, *Huffman*. Figure 5.8(a) shows that the Spearman's rho value for Drupal rank is 0.48, custom rank is 0.59, pageview-based rank is 0.39, revised rank is 0.54, and DSN-based rank is 0.4. Thus, according to Spearman's rho, custom rank performs the best for query term *Huffman* and pageview-based rank performs the worst (largest p-value of the five ranks). The result from Kendall's tau is slightly different where revised rank has the highest τ value (0.45) and pageview-based rank has the lowest



(a) Rank correlation coefficients

(b) p-values of rank correlation coefficients

Figure 5.7: Rank correlation for query term *Merge Sort*.

τ value (0.23). The last evaluation, Pearson's coefficient, produces values similar to Spearman's rho.

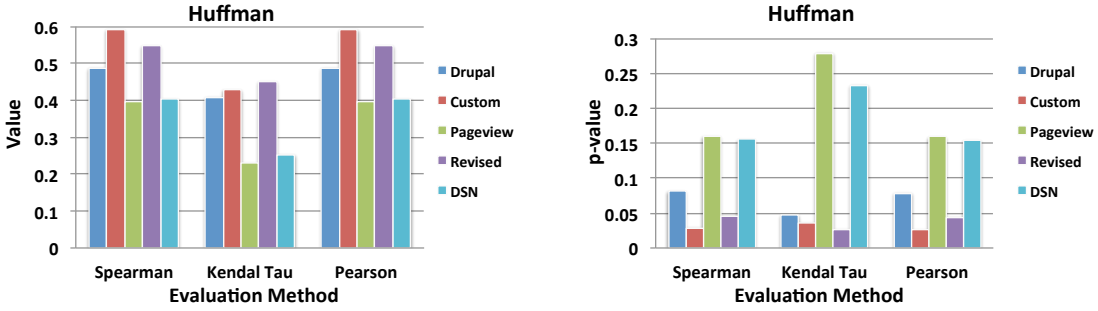
The p-values in Figure 5.8(b) show that according to Spearman's rho and Pearson's coefficient, both custom rank and revised rank exhibit significant correlation with the log-based rank. The p-value for the coefficients are less than 0.05 for both these ranks for both of the rank correlation coefficients. All of Drupal rank, pageview-based rank, and DSN-based rank have p-values greater than 0.05. Kendall's tau shows a similar trend except it shows that the Drupal rank is also significant.

With query term *b tree* in Figure 5.9(a) we see that two out of the five ranks have negative values (i.e., Drupal and custom rank). The results are similar for all of the evaluation methods, except that the values of Kendall's tau are smaller compared to Spearman's rho or Pearson's coefficient. Figure 5.9(b) lists the corresponding p-values which show that both pageview-based rank and DSN-based rank are significantly correlated with log-based rank.

Lastly, with the query term *quick sort* in Figure 5.10(a) the largest Spearman's rho value is generated by Drupal rank (0.32). All of custom rank, pageview-based rank, and DSN-based rank produce negative ρ values. However, revised rank results in positive ρ values. Similar trends can be seen for both Kendall's tau and Pearson's correlation coefficient. Figure 5.10(b) shows the p-values for the coefficients. As we see from this figure, any of the five ranks fail to show any significant correlation with log-based rank.

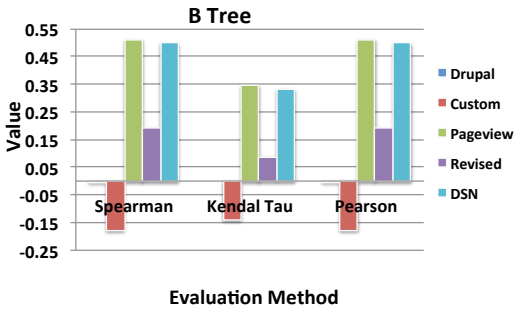
In this section we tested five ranks with ten queries against the log-based rank. Table 5.2 shows the number of significant results for each of the five ranks. As the table shows, the Drupal rank fails to show any significant correlation with log-based rank in each case we tested. Compared to Drupal rank, custom rank was able to show significant results in two out of the ten cases (i.e., Merge sort and Huffman).

Overall, the pageview-based rank and DSN-based rank show significant correlation with log-based rank for most cases (i.e., six out of the ten cases we tested). The pageview-based rank fails to show a significant result with query terms *quick sort, heap, bubble sort*, and *Huffman*. Pageview is a widely used standard for identifying user interest. However, in our case, there are a number of caveats of solely using pageview as a measure for ranking. Pageviews can be generated by a number of different activities that include browsing, searching, following a link in email or a webpage, etc.

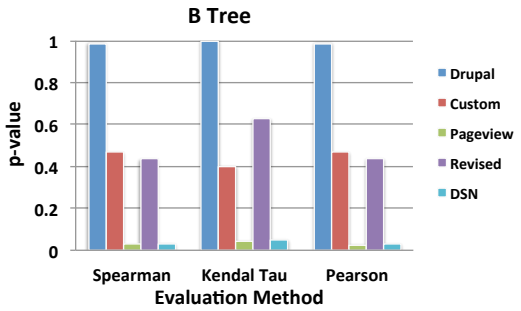


(a) Rank correlation coefficients (b) p-values of rank correlation coefficients

Figure 5.8: Rank correlation for query term *Huffman*.



(a) Rank correlation coefficients



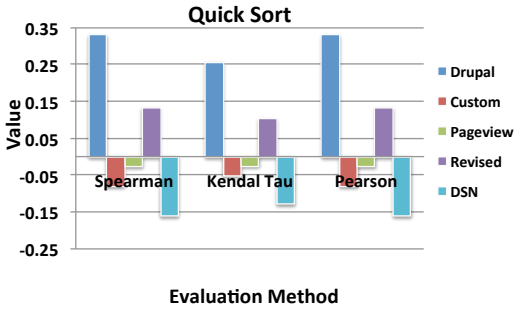
(b) p-values of rank correlation coefficients

Figure 5.9: Rank correlation for query term *B Tree*.

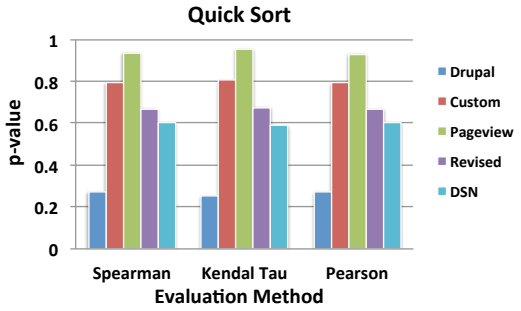
With the current logging system, it is not possible to identify activity-specific pageview. In an ideal situation, a search-specific pageview can give us insight into which pages are looked at by users during a search. Unfortunately, our current log data collection does not capture this information. Another drawback of solely using pageview in ranking is that it fails to include the custom fields that may contain potentially useful information.

Out of the ten cases, revised rank was able to show a significant result eight times. It failed to show significant results for *b tree* and *quick sort*. However, with *quick sort* none of the schemes are able to show any significant correlation with the log-based rank. With *b-tree* the performance of revised rank is affected by custom score. While computing revised score (see Equation 5.5) we do not place any weight on the custom score or on any of the coefficients (social interest coefficient, and weighted interest). We believe introducing a weighting factor on the custom score while computing the revised score will allow us to tune the effect of custom score on the overall rank.

Of the five ranks we tested, custom rank includes custom fields or domain specific knowledge. Incorporating domain specific knowledge into various DL services is important for better serving the user of a certain DL. Custom fields are specific to AlgoViz catalog entries and can be beneficial in identifying potentially useful AV entries. Although custom rank uses custom fields, this ranking fails to successfully reflect user trends in most cases (it fails in eight out of the ten cases we tested). Drupal ranking on the other hand tries to combine user interest (number of views in Table 5.1)



(a) Rank correlation coefficients



(b) p-values of rank correlation coefficients

Figure 5.10: Rank correlation for query term *Quick Sort*.

Table 5.2: Significant rank correlation for different ranking.

Ranking	# of significant correlations	Query terms with significant correlations
Drupal	0	-
Custom	2	Huffman, Merge sort
Pageview	6	Binary search, AVL, Stack and Queue, Dijkstra, Merge sort, B-tree
DSN	6	Binary search, AVL, Stack and Queue, Dijkstra, Merge sort, B-tree
Revised	8	Binary search, AVL, Stack and queue, Dijkstra, Heap, Bubble sort, Merge sort, Huffman

with content-based ranking by using the title and body of the resource while ranking. Note that Drupal ranking does not include the custom fields of resources into the ranking system. This ranking uses number of comments and the recency of comments as indicators of user interest. While these indicators do show the user interest in a resource, edu-DLs that suffer from lack of user activities will not particularly benefit from such indicators.

Pageview-based rank solely depends on pageviews which can be generated by a number of different activities. Some of these activities may pose bias towards a certain resource, such as a website containing the URL of a particular resource. Using only one form of user activity to rank content poses a significant risk of producing ranks that are not useful. Besides, pageview-based ranking also does not include domain specific knowledge. Similar to pageview-based rank, DSN-based ranking solely depends on user activities. However, in certain cases DSN-based rank shows higher rank correlation coefficients (e.g., AVL) or lower p-values (e.g., Huffman, quick sort) compared to the pageview-based rank.

Revised ranking, which builds upon custom ranking and DSN-based ranking, provides a hybrid approach. Note that we do not use the information of the isolated users in revised ranking. Machine learning algorithms can be used to build a model for effectively ranking content with different DSN-derived information including the isolated users. Revised ranking uses both domain knowledge of the resources (e.g., custom fields) and user trends to rank content. The two DSN-derived coefficients used in revised ranking capture both user interest on a resource across different user groups and the weight of that interest. In our comparisons, revised rank shows the most promising result while combining both domain knowledge and user interest. We believe edu-DLs that lack explicit user activities (e.g., comments, ratings) can benefit from including the DSN-derived information into the ranking system along with the content-based information.

5.4 Application architecture

Edu-DLs that lack active user participation and use traditional search services can benefit from incorporating DSN-derived information into the ranking of search results. In this section we briefly describe an architecture that includes DSN-derived knowledge into the ranking system. Figure 5.11 shows the framework. Two offline activities form the core of this framework. The first offline module is the Indexer Module that indexes the content. Indexing is a core part in any information retrieval

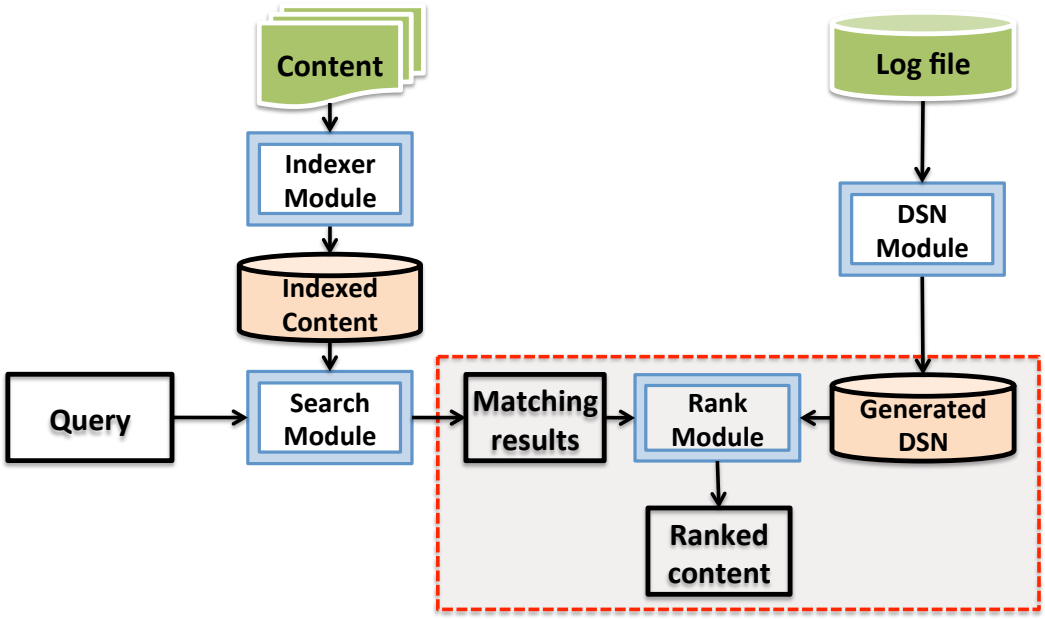


Figure 5.11: Ranking resources during search using DSN.

system. The DSN Module depends on log data to track user activities. This module cleans the log data, generates the deduced social network, and partitions the network. All three activities of the DSN module can be done offline.

The search Module takes a user query and indexed content to produce a set of matching results. The results then pass through the Rank module which uses metadata of the matching results as well as the groups in the DSN to generate the ranked results.

The DSN Module takes a number of parameters from the administrator such as the type of the DSN (e.g., user-user or page-page), the time period for which the DSN should be built, the connection threshold, the partitioning algorithm, etc. The ideal parameters of any edu-DL will depend on the type of the DL and the type of user activity in that DL.

5.5 Summary

In this chapter we present a revised ranking system for AlgoViz. This system uses the existing AlgoViz ranking combined with the information retrieved from the DSN to include usage information. Our evaluation shows that there is significant correlation between the revised rank and log-based rank (an objective measure of demonstrated user interest in the search results). That is, revised ranking performed better than the existing custom ranking alone. This approach can be used in other DLs where user modeling is difficult yet social navigation would be a valuable addition.

Chapter 6

Recommendation in an Educational DL

Educational digital libraries connect content with users. Groups might be implicitly formed within these libraries when similar users, such as educators, come here and share common actions. The vast amount of resources and activities within an edu-DL lead to the problem of information overload — how to find useful resources in a reasonable amount of time when there is too much information to manage. In many edu-DLs the prevalent practices within the groups are not visible. Users act (e.g., browse, search) mostly on their own and lack any knowledge of how other users acted in a similar context.

Capturing user activities within an edu-DL can shed light on the potential usefulness of a resource. Making practices visible not only can show users what others are doing but also can motivate users to explore content and share their experience. DSNs can be useful in such scenarios since they capture user activities. In this chapter, we present a DSN-dependent recommendation framework that can be used to recommend content based on user activities within an educational DL.

6.1 Model building based on a DSN

Recommendation systems are widely used for easing information overload [5, 24, 131]. These systems not only capture, store, and analyze user activities, they also present popular trends to users. These systems can be grouped into three broad categories: content-based recommendation, collaborative recommendation, and hybrid recommendation (see Section 2.4.4).

In content-based recommendation, the similarity between content is the leading factor for recommending like resources [107]. The similarity measure can differ depending on the type of the content. For example, a system with text corpora will require textual similarity to find closely matching documents. Some e-commerce sites use categorization of commodities for similarity measurement [134, 123].

Collaborative recommendation, however, depends on similarity calculations based on user profiles and activities [48, 24]. There should be a considerable number of users with accounts and activities (e.g., rating, review) in the system for collaborative recommendation to perform well. The user

accounts and activities are used to build user models which are later used to recommend resources to a new or existing user. The last type of recommendation, hybrid-recommendation, combines content-based recommendation and collaborative recommendation.

Many edu-DLs are rich in user activities yet typically lack user information such as accounts, ratings, comments, etc. In the absence of such explicit user activities, DSNs can provide a better insight into existing user trends. DSN-based recommendation, when used with text-based recommendation, would result in a hybrid recommendation system that uses the DSN derived information to generate a model to recommend content.

As shown in Figure 4.2, we use the accesslog table of AlgoViz to create a DSN. The DSN-building step takes two parameters: the time range when the network is built (e.g., Fall 2010, January 2013) and the connection threshold k. These parameters control the size and connectivity of the resultant network. A longer time range keeps more log data and produces a larger network. Smaller connection threshold results in more connectivity and creates a denser network.

The constructed DSN becomes an input for a network partitioning module that uses modularity clustering to detect groups in networks. Resultant groups $g_1, g_2, \ldots, g_n$, are used in the recommendation phase. In the recommendation phase, we build a logistic regression model to estimate a score for a pair of pages that might be seen in a particular user session. The model takes content based similarity, user's group information, and user activity within his/her group and across other groups. Finally, we utilize the outcome of this model to recommend resources to the user. The following subsections describe the methodology.

6.1.1 Logistic regression

Our objective is to recommend content to a user based on which group in the DSN is most similar to that user. Our observation is that the group-based recommendation system we have developed performs better in identifying user interest than text-based recommendation. The groups identified from the DSN bring users with similar interest together in the same partition. Resources used by peers of the same group are considered to be candidate elements for the recommendation. We design a logistic regression model to capture a similarity score between a pair of resources used by a user.

Consider a situation where each group g_i contains a set of users $u_1, u_2, \ldots, u_j$. Each user has at least one but possibly multiple sessions. Each session contains pages, $P = p_1, p_2, \ldots, p_k$ visited during that session. The session information can be used to create objective data, also known as ground truth, to train the model. We model the binary ground truth, whether a pair of pages p_x and p_y exists (or not) for a user u_j in the log data, in one equation (see Equation 6.1) using content similarity, the frequency of p_x and p_y, and information collected from the user's group g_k. This is based on our assumption that two pages are likely to be related if they appear in the same session. Our hypothesis is that membership in a group is an important consideration for a recommendation model. The question we try to answer is: **Given that page p_i was seen by user u_j who belongs to group g_k, how likely is that u_j will view page p_n?**

The use of content information alone in the modeling cannot capture the community dynamics in the recommendation. Dependence on user trends entirely, on the other hand, can be risky, especially when user activities are scarce. The proposed recommendation model uses two types of information

that reflect its hybrid nature: text similarity and user activity. The log table we use contains the title of the page and URL — both of which are used as text information for a resource. The DSNs on the other hand will provide us with collaborative information.

To model the existing binary information, whether a user u_j of group g_k has seen pages p_i and p_n in the same session, we compute the following:

$$\begin{aligned} c_1 \times \text{Similarity between titles } (p_i, p_n) + \\ c_2 \times \text{Longest common prefix } (LCP) \text{ in URLs } (p_i, p_n) + \\ c_3 \times l(p_i, p_n) + \\ c_4 \times m(p_i, p_n, g_k) \end{aligned} \tag{6.1}$$

where $c_1, c_2, \ldots, c_4$ are the model coefficients, $l(p_i, p_n)$ is the number of times (p_i, p_n) is found in sessions in the DSN, and

$$m(p_i, p_n, g_k) = \frac{\#\text{ of times } (p_i, p_n) \text{ appears in sessions in } g_k}{\#\text{ of users in } g_k}. \tag{6.2}$$

The first two terms in this model capture the text similarity and the last two terms represent group information. If a certain page pair does not appear in a DSN, this model will use text-based information to find similarity between the pages. When the page pair is present in the DSN (that is, at least two users viewed these pages together in some sessions), the model includes the group information through l and m. While l provides an idea about the general user interest in a page pair across all the groups in the DSN, m shows the popularity of a page pair in a certain group. In particular, m represents the average number of times each user of the group viewed this page pair.

While the values of *l, m, LCP*, and *similarity* can be computed from the data, we need to build a model that will provide the values for the coefficients. Building a good model will result in a set of coefficients that is able to best capture the binary outcome of viewing two pages in a session. Probabilistic modeling or optimization algorithms are a few techniques for building a model to find the best coefficients for each parameter in an equation [103, 98, 99]. We choose to use logistic regression [102] to build the model. It is a probabilistic classification method that performs well in situations where the dependent variable is binary. That is, the outcome always will be either zero or one, success or failure. The predictor (independent) variables in logistic regression can take values which may be discrete, continuous, or categorical, and do not need to be correlated. Logistic regression is particularly suitable for our problem because we have different types of uncorrelated data.

A logistic regression uses logarithms and takes on a similar approach to linear regression. It uses a logistic function which can have a outcome between zero and one.

$$F(x) = \frac{e^x}{e^1 + 1} = \frac{1}{1 + e^{-x}} \tag{6.3}$$

where e denotes the exponential function and x is the linear function we described in Equation 6.1. Since x is a linear function of more than one predictor variables, then Equation 6.3 becomes

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{e^{\beta_0+\beta_1 x}+1} = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \tag{6.4}$$

where β_0 is the intercept of the underlying linear regression model and β_1 is the regression coefficient for predictor variable x. Note that the value of $\pi(x)$ will range from zero to one.

The inverse of the logistic function is called *logit*, which is defined as

$$\begin{aligned} g(x) = \ln \frac{\pi(x)}{1-\pi(x)} &= \beta_0 + \beta_1 x \\ \text{or, } \frac{\pi(x)}{1-\pi(x)} &= e^{\beta_0+\beta_1 x} \end{aligned} \tag{6.5}$$

where ln is the natural logarithm.

Logistic regression is particularly suitable for our case because $\pi(x)$ always returns a value between zero and one, which can be converted to conditional probability of class membership (i.e., pair exists/does not exist). To rank the recommendations for a particular page p_i seen by user u_j, we can directly order pages p_n based on the conditional probability of existence of the pairs (p_i, p_n) where p_n can be any page other than p_i.

For our case, logistic regression requires both positive examples (existence of a page pair in a session) and negative examples (absence of a page pair in any session). We generated pairwise positive examples from the log data. We created the negative examples by altering the positive data set and randomly generating pairs and validating their absence in the log data. We used the LingPipe API [8] to implement logistic regression. LingPipe is a widely used text processing toolkit that uses computational linguistics. The LingPipe API for logistic regression provides an option for multinomial classification. In order to train a logistic regression model, LingPipe requires the inputs to be represented using a matrix and outputs as integers. We used the default values for the other parameters such as *RegressionPrior.noninformative()* or 0.000000001 as the minimum improvement value for the search for the estimate.

6.1.2 Resource pair proximity model

In this chapter we use two AlgoViz DSNs created from the log data of Fall 2009 and Spring 2010. The connection threshold is 10. Partitioning the networks resulted in the detection of six groups in the Fall 2009 DSN and 12 groups in the Spring 2010 DSN. Figure 6.1 shows the number of users for each cluster detected in these DSNs. Both DSNs have three groups of users with more than 100 users. However, the number of groups with fewer users are more in the Spring 2010 DSN, indicating the presence of more users with diverse interest. Note that the number of users in the Spring 2010 DSN is twice the number of users in the Fall 2009 DSN (see Figure 4.4). The increase in users could be a reason for having more groups in the Spring 2010 DSN.

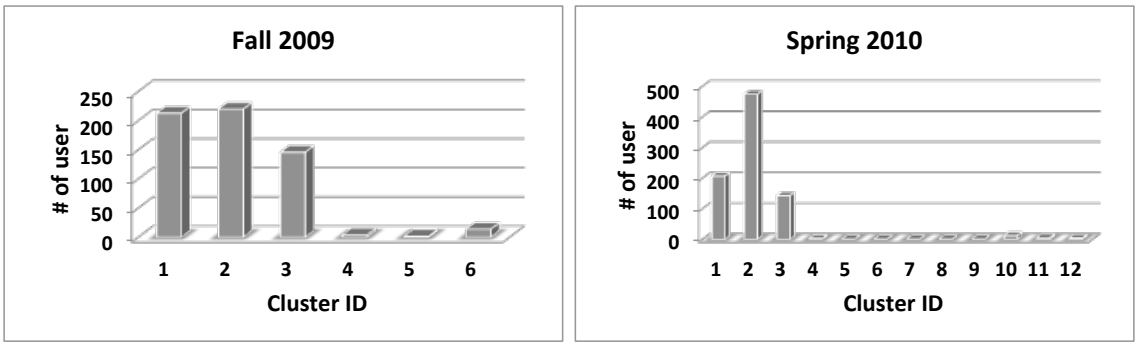


Figure 6.1: Number of users in the groups found in the DSNs.

Figure 6.2 describes the methodology for building a model using logistic regression. Logistic regression needs both positive and negative examples to build a model. We use the log data to build the positive examples for the Model Builder module in Figure 6.2. The cleaned log file that was used to generate the DSN is used at this stage to generate all pairs of pages, (p_x, p_y) that appear in any session. For each pair (p_x, p_y), we compute several values. Parameter l stores the number of sessions that contain (p_x, p_y). It acts as a global indicator of how frequently a pair of pages appears together in sessions regardless of any group. Group information is considered in parameter m which is computed by dividing the number of times a pair of pages appear together in any session in a given group g_k by the number of users in g_k (see Equation 7.5). For a given group, this parameter provides information on how many times on an average a user viewed this page pair. In most cases, the denominator helps offset the effect of large groups or the most active users within a large group.

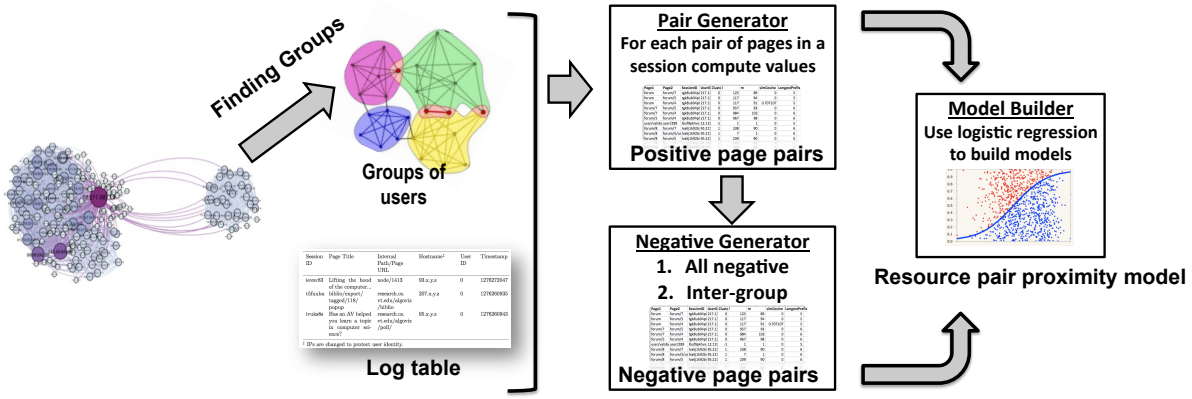


Figure 6.2: Recommending resources using a deduced social network.

We compute the similarity between the titles of pages in pair (p_x, p_y) using the Cosine similarity measure [42]. The accesslog table also contains the page URL along with the page title. We include this information in the model equation under parameter LCP. We use a different measure to identify if two pages are of the same type, based on the URLs. Instead of measuring the similarity between URLs, in this case we compute the longest common prefix (LCP) of the URLs. Usually, the greater the LCP, the closer the page type. For example, *forum/7* and *forum/4* are of type *forum* while

taxonomy/term/4/ and *taxonomy/term/5/* are of type *taxonomy* although they differ in *taxonomy* terms.

The page pairs and computed information such as l and m form the positive examples for the logistic regression. In order to build a good model we also need negative examples. Since the accesslog table only contains pages visited in any given session, we must generate the negative examples where a pair of pages was not visited in a session. The Negative Generator module of Figure 6.2 attempts to create an equal number of negative examples of two types.

1. **All negative examples:** Select a pair of pages (p_x, p_y) such that (p_x, p_i) and (p_i, p_y) appear in some sessions but (p_x, p_y) does not. Having a common page p_i in two pairs of pages indicates that pages p_x and p_y may share something in common. We use (p_x, p_y) as a negative example. The values of parameter l and m are both one since this pair of pages (p_x, p_y) does not appear in any session in any group.

2. **Inter-group negative examples:** For any given group g_k, select a pair of pages (p_x, p_y) such that it appears in session $sess_{i\_g_k}$ but does not appear in any session $sess_{j\_g_m}$ in group g_m. This page pair from group g_k is then used as a negative example for group g_m. Only the value of parameter l, computed for (p_x, p_y) in group g_k, remains the same. The value of m is one for group g_m.

Once we have computed the positive and negative examples, we use logistic regression to build a model for page pairs using both example sets. We call the resultant model a *resource pair proximity model*. This model provides an estimation of how likely a pair of pages will appear together in a session for a user belonging to a particular group.

6.2 Recommending resources

Figure 6.3 shows the framework used to recommend content based on the classifiers. We begin with the resource pair proximity model and a list of pages $pt_1, pt_2, \ldots, pt_m$ which were used to train and build this model.

The resource pair proximity model depends on four terms, *l, m, LCP*, and *similarity* and the model coefficients $c_1, c_2, \ldots, c_4$, to estimate the probability of viewing two pages in a session. In the absence of group information (i.e., m) this model depends on the other DSN-derived parameter l. When both of these parameters are null it indicates that the given pair of pages was not visited by any user in any session. In this case, the resource pair proximity model depends on the title and URL similarity.

The resource pair proximity model provides us with the coefficients for Equation 6.1. In order to estimate if a pair of pages is likely to appear together in a session we also need the values of l and m computed from the log data. While l is a common parameter across all groups, m is a group-specific parameter. When a user starts viewing pages in AlgoViz, a session is automatically created that logs the pages visited. To recommend pages to this user using the resource pair proximity model,

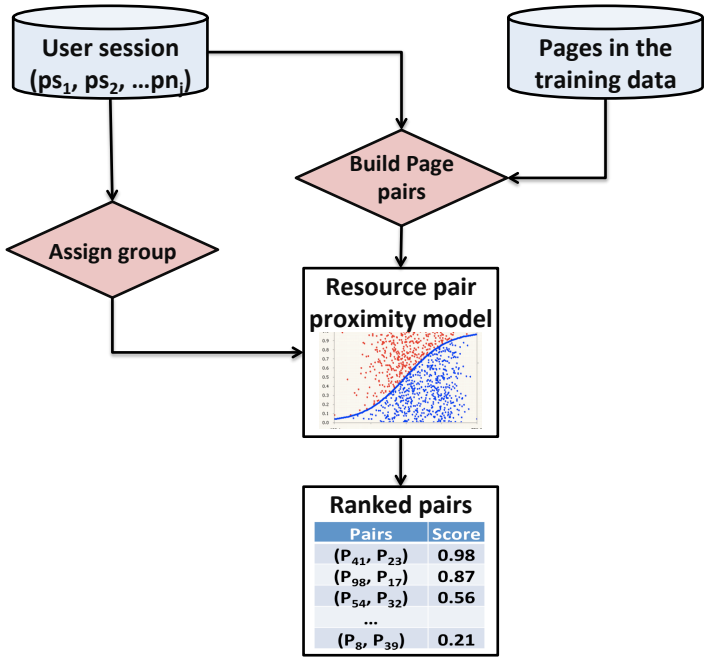


Figure 6.3: Recommending content based on the Resource pair proximity model.

we assign this user to a group, as described next. This assignment will allow us to compute the values for group-dependent parameter m.

Clustering algorithms are often used to assign a user to a group [63]. Approaches such as point-based measures or cluster centroids are often used to assign newly arriving points to an existing cluster. We use the centroid-based approach since it is a popular scheme for compact clusters which are similar to the clusters we see in the AlgoViz DSN [63]. In order to assign a user to a group, we first compute group centroids. Group centroids are computed by taking the average of all the weights of the various terms present in a page pg (i.e., title and URL) visited by the group members. Given a set of pages and their corresponding vector representations pv, the centroid vector gc is defined as

$$gc = \frac{1}{pv} \sum_{pg \in pv} pg \tag{6.6}$$

A similar measure is used to find the centroid of a user. We select the pages visited by the user and use the page vectors to compute user centroid uc. After the centroids are computed we use Euclidean distance to measure the distance between the user and the groups. Euclidean distance between two vectors is computed as:

$$dis(gc, uc) = \sqrt{(gc_1 - uc_1)^2 + (gc_2 - uc_2)^2 + \cdots + (gc_n - uc_n)^2} \tag{6.7}$$

where n is the length of the vectors.

The user is then assigned to the group that has the least distance from the user centroid. Once a

user u_j is assigned to a group, we take pages $p_1, p_2, \ldots, p_x$ from his session and pages viewed by other users $po_1, po_2, \ldots, po_y$ to create all possible page pairs (p_x, po_y). One page of (p_x, po_y) comes from the session of u_j and the other page comes from the sessions of other users (it also may appear in u_j's session). We compute the estimated probability of viewing these pages together in a session for all these page pairs following Equation 6.1. Since the values range from zero to one, they can be used to rank the pages. This ranked list of pages then can be used to recommend potentially useful content to the user.

Assigning a user to a group can be done both online and off-line. When we generate the DSN for AlgoViz, there are a number of users who are not part of the DSN. Figure 4.4 shows the number of isolated nodes (i.e., users who are not connected to the DSN) in different datasets. As we see from Figure 4.4, the number of isolated nodes in the DSN is high compared to the number of connected nodes (e.g., 2224 components vs. 203 connected nodes in Fall 2009). One reason this might happen is because the connection threshold k is set higher than the total number of pages these isolated users visited. Another possibility is that the number of common pages between any of these isolated nodes and any other connected user is less than k. However, while building the resource pair proximity model we include the information of these isolated users in the model through the predictor variable l. Thus, if an isolated user viewed a page pair (p_x, po_y) in a session, the l value of (p_x, po_y) will be increased by one. The value of m for this page pair of the isolated user, however, will be zero since the isolated user does not belong to a group.

6.3 Evaluation

The recommendation framework that we presented in this chapter has a number of components. We carry out separate evaluations for different parts of this framework. We begin the evaluation with analysis of the goodness of the model built using logistic regression. We then test the accuracy of the classifier that uses the model, and lastly test the accuracy of the recommendation provided by the recommendation system that uses the classifier. Details of each of these evaluation methods and results are given in the following subsections.

While evaluating, we use the n-fold cross validation technique [110]. We started with 10-fold where the data was divided into ten equal segments (i.e., fold). One segment was used for testing and the other nine were used to build the model. We iterated over the folds such that each fold becomes the testing segment at some iteration. The average results from the test segments are plotted in the figures. For a given fold, we take the average of all the results. For example, at 6-fold, the data is partitioned to six segments, five of which are used for training and the remaining one for testing. We iterate over the segments six times such that a separate segment becomes the testing fold at each iteration. After the sixth iteration we compute the average which is plotted as the final result for 6-fold.

Most often an 80-20 or 90-10 split is used for training-testing. For each evaluation we started from 10-fold and went down to 6-fold. This means that we decrease the amount of training data when moving from 10 fold to 6 fold. At 6-fold, each fold contains 16.6% of total data. Thus, at this fold, around 84% of the total data is used for training and 16% is used for testing. By closely varying the folds we are able to better track the change in performance.

To perform the evaluations we built four models with different predictor variables: S, PS, LPS, and $MLPS$. We wanted to compare the performance of text-based models with models that include DSN-derived information. Of the four models we tested, the first two depend on text similarity, and the last two include DSN-derived information with text similarity. While the first two models are similar to content-based recommendation (see Section 2.4.2), the last two models represent the hybrid recommendation approach (see Section 2.4.4).

The first model, S, only contains the similarity variable in the model equation. The PS model contains both S and the longest common subsequence (LCP). The next model, LPS, contains the DSN-dependent parameter l along with the previous parameters. Information about isolated users is included in this model. Thus, if an isolated user viewed a page pair in a session, the value of l for that particular page pair will include this information. The last model, $MLPS$, includes the DSN-based group parameter m with the three other predictor variables. Note that since isolated users do not belong to any group, parameter m does not include information about isolated users. However, since l contains the session information for isolated users, both the LPS and $MLPS$ models contain information about isolated users.

For each type of evaluation method we tested the four models under various folds. Table 6.1 shows one sample of the coefficients for the four models built using the Fall 2009 DSN at 10 fold.

Table 6.1: Coefficients of the model for Fall 2009 DSN.

Model	Intercept	c1 (M)	c2 (L)	c3 (P)	c4 (S)
MLPS	-9.7445147	3473.78307	-0.0005	-0.004251	-4.438316
LPS	-24.152219		0.14045626	-3.864488	-0.830244
PS	-2.6033178			-1.1006378	0.28259037
S	-3.3611924				0.00437504

6.3.1 Model evaluation: goodness of fit

We began our evaluation with testing the model that is built using logistic regression. Although logistic regression is somewhat similar to linear regression, classic regression analysis evaluation methods (e.g., variance, chi-square test) are not suitable when it comes to testing the model and the goodness of its fit. Since logistic regression uses a log function, the resultant model usually lacks a typical linear trend. As a result, for models built using logistic regression, a number of different evaluation methods such as deviance, $Pseudo - R^2$, *Cox and Snell* R^2, etc. are used to measure the fit of the model.

Deviance is similar to residual sum of squares (RSS) of a linear model [4, 57]. Deviance is computed using log likelihood of a model. It is defined as:

$$D_{model} = -2 \log L_{model} \tag{6.8}$$

where L is likelihood.

When building a model, logistic regression tries to minimize the deviance. Deviance itself does not indicate the quality of a model. To test the goodness of fit for a logistic regression model, usually

the deviance of the model is compared with the null deviance. Null deviance is the deviance of a model without the predictor variables. Pseudo-R^2 is such a measure for testing the goodness of a fit using deviance [76]. It is defined as:

$$\text{Pseudo-}R^2\text{s} = \frac{D_{\text{null}} - D_{\text{model}}}{D_{\text{null}}} \tag{6.9}$$

The denominator of the ratio is the deviance from the null model. The numerator of the ratio is the difference between the deviance of the fitted model and the null model. The ratio indicates the improvement caused by inclusion of the model parameters in the null model. A value of Pseudo-R^2 closer to one indicates greater improvement caused by the fitted model as compared to the null model.

Two other similar measures that we use to test the goodness of the fit are Cox-Snell R^2 [28] and NagelKerke R^2 [92]. The Cox and Snell R^2 is defined as:

$$f(x) = 1 - exp\{\frac{-2(LL_{fitted} - LL_{null})}{N}\} \tag{6.10}$$

where LL_{fitted} is the log likelihood of the fitted model, LL_{null} is the log likelihood of the null model, and N is the sample size.

Following the Cox-Snell R^2 measure, an outcome closer to one indicates a good fit. NagelKerke R^2, which is similar to the Cox and Snell R^2, is also used to test the goodness of fit of a model built using logistic regression. The NagelKerke-R^2 equation is defined as:

$$f(x) = \frac{1 - exp\{\frac{-2(LL_{fitted} - LL_{null})}{N}\}}{1 - exp\{\frac{2LL_{null}}{N}\}} \tag{6.11}$$

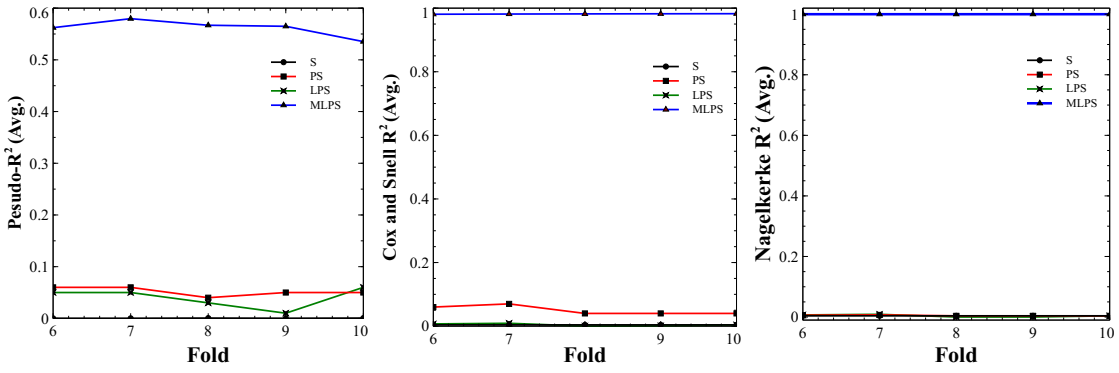


Figure 6.4: Evaluation of the model built using AlgoViz Fall 2009 data.

Figure 6.4 (left) shows the Pseudo-R^2 values for different models at different folds. These models are built using the AlgoViz Fall 2009 dataset. The X axis shows the number of folds and the Y axis shows the average Pseudo-R^2 value. Each line in this plot represents a different model with different numbers of predictor variables. This plot shows that according to the Pseudo-R^2 measure,

the complete model ($MLPS$), represented by the blue line, performs better compared to the other three models. The model with S, denoted by the black line, has the worst performance which is close to zero. The LPS model, denoted by the green line, shows slightly improved performance. Similar results are obtained for the PS model. All three models without the M variable have Pseudo-R^2 values that are closer to zero, meaning that there is no significant improvement of the model when it is compared with the null model - the model without any predictor variable. A good model will result in a Pseudo-R^2 value closer to 1. Out of the four models, only $MLPS$ results in values much greater than zero. The value of $MLPS$ ranges between 0.4 to 0.6 in this figure. Thus it shows that models built with group information derived from the DSN perform better compared to content-based models in predicting the likelihood of viewing two pages together in a session.

Figure 6.4 (center) presents the Cox and Snell R^2 values for different models at different folds. According to this method, a value close to one indicates a good model. For the Fall 2009 AlgoViz dataset, only model $MLPS$ produces a value which is close to one. Models PS and LPS both produce very small values which are slightly greater than zero. Model S produces values closer to zero. Figure 6.4 (right) shows the NagelKerke-R^2 values for different models. As we stated earlier, this measure is similar to the Cox and Snell measure. Thus this plot is similar to the center plot.

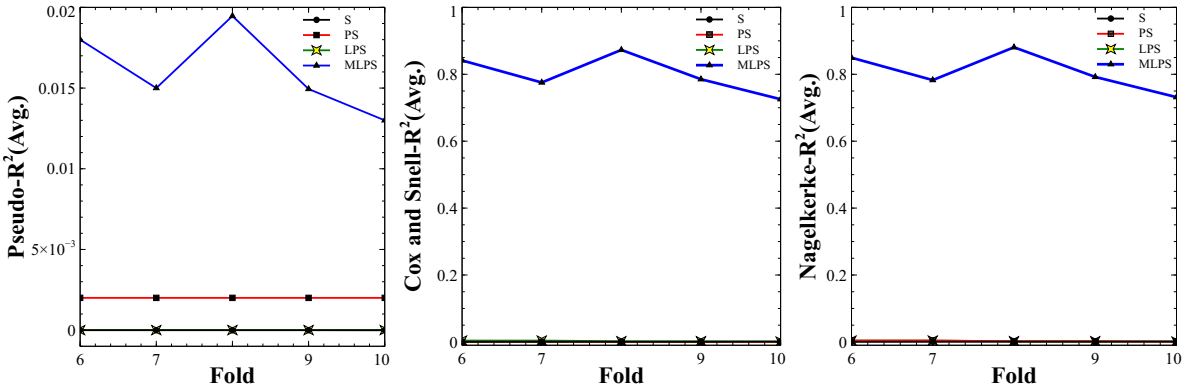


Figure 6.5: Evaluation of the model built using AlgoViz Spring 2010 data.

We perform a similar set of evaluations with AlgoViz Spring 2010 data. Figure 6.5 (left) shows that the highest Pseudo-R^2 value is achieved by the $MLPS$ model. Models S and LPS overlap each other and are close to zero indicating neither of them can provide a good model. Model PS performs better than the last two models but not as well as the $MLPS$ model. Figure 6.5 (center) shows the Cox and Snell R^2 measure. This figure also indicates the $MLPS$ model performs better compared to the other three models. Similar results are seen when we use the NagelKerke-R^2 approach (see Figure 6.5 (right)).

As we see in these figures the model with all four predictors performs better than the other models. While the inclusion of l (one of the two DSN-dependent variables) does not provide significant improvement, including the other DSN-dependent variable, m, provides a better measure for the goodness of fit. From this analysis we can conclude that the group-specific variable m is important in building a model following Equation 6.1.

One point to note here is that the Pseudo-R^2 values for the $MLPS$ model for the Spring 2010 DSN

are much lower compared to the Fall 2009 DSN. However, the other two R^2 values are much higher compared to Pseudo-R^2 values for the Spring 2010 DSN. One of the drawbacks of Pseudo-R^2 is that the rate of change in Pseudo-R^2 is not proportional to the odds ratio. This could contribute to the lower Pseudo-R^2 value, yet higher Cox and Snell R^2 and NagelKerke-R^2 values.

6.3.2 Classifier accuracy

Once the resource pair proximity model is generated it is used to estimate the probability that a pair of pages appear together in a session. Based on this estimated value we can predict whether both the pages of a pair appear in the same session or not. Thus in this case the model would act as a classifier. We conducted another set of evaluations to test the performance of the classifier. As we mentioned earlier we used the n-fold cross validation technique. After a model is trained with the training folds, we perform the test with the testing fold. The model can produce four different outcomes for an instance of the test data:

1. True Positive (TP): The model predicts 1 when the outcome should be 1.
2. True Negative (TN): The model predicts 0 when the outcome should be 0.
3. False Positive (FP): The model predicts 1 when the outcome should be 0.
4. False Negative (FN): The model predicts 0 when the outcome should be 1.

With these measures, the accuracy of a classifier [108] can be computed as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.12}$$

Accuracy shows the fraction of correct predictions. The fraction of correct positive predictions is known as Precision [108] which is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6.13}$$

There are two other measures: recall and specificity, that are widely used to test classifier performance. Recall or sensitivity [108, 9] provides the fraction of positive cases that are predicted as positive. Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{6.14}$$

Specificity [9] on the other hand provides the fraction of negative cases that are predicted as negative. It is defined as:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{6.15}$$

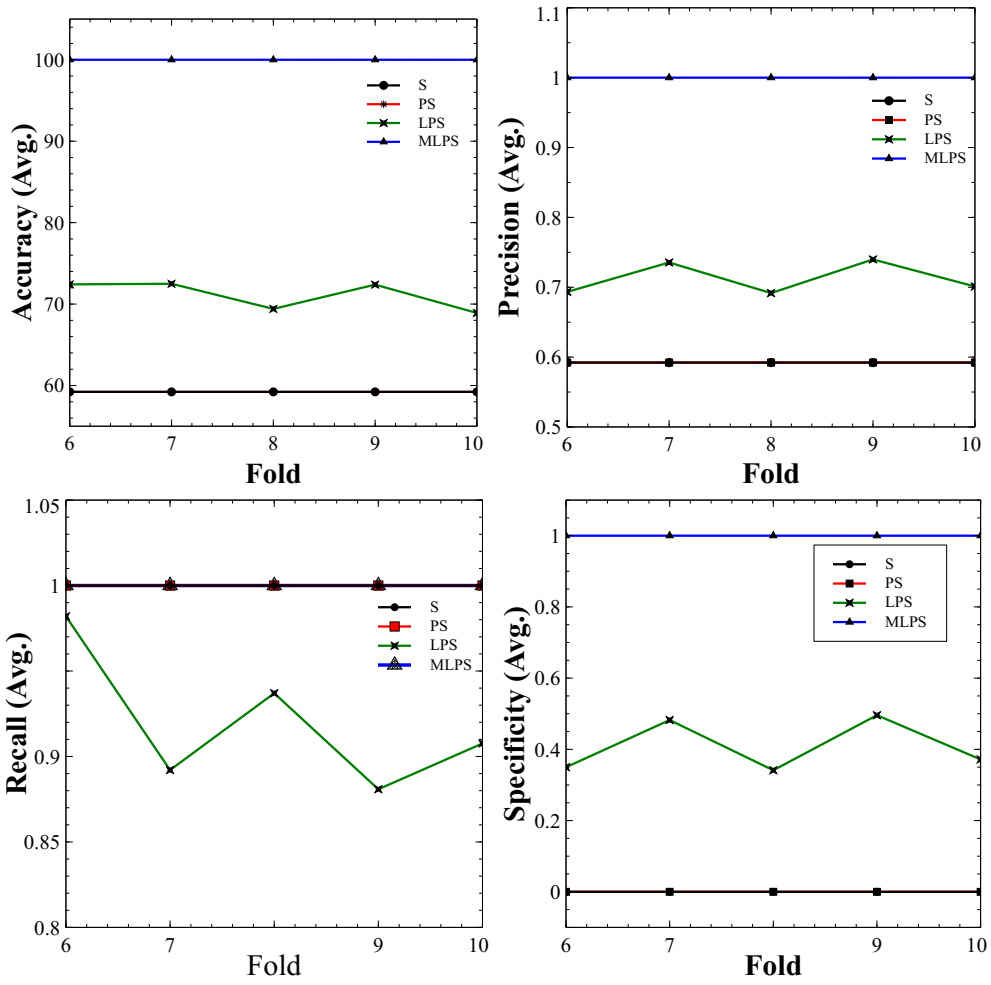


Figure 6.6: Classifier evaluation using the models in Figure 6.4 (AlgoViz Fall 2009 DSN).

We use two AlgoViz datasets to evaluate the classifier. The classifier for each dataset is tested using four different methods: accuracy, precision, recall, and specificity. Figure 6.6 and Figure 6.7 show the experiments results. The first plot in both figures (i.e., accuracy) uses a percentage scale in the Y axis. The other three plots in the figures use a scale of 0 to 1 in the Y axis. While describing the results for these three plots we will use percentages by converting the fractions.

Figure 6.6 (top left) shows the accuracy of the classifier for the AlgoViz Fall 2009 dataset. As we see when only similarity (i.e., Cosine similarity in this instance) between a pair of pages is used to create the model (S), the average accuracy is around 60%. The accuracy of the model with PS is also similar to S, hence the lines overlap. Model LPS (the green line) shows an accuracy slightly over 70%. Considering longest common prefix (P) and l along with the similarity increases the accuracy to around 70-80%. Again, l is a global indicator of how likely two pages are to appear together in a session. Once we include the local parameter m into the model along with the existing parameters, the average accuracy jumps to around 99%.

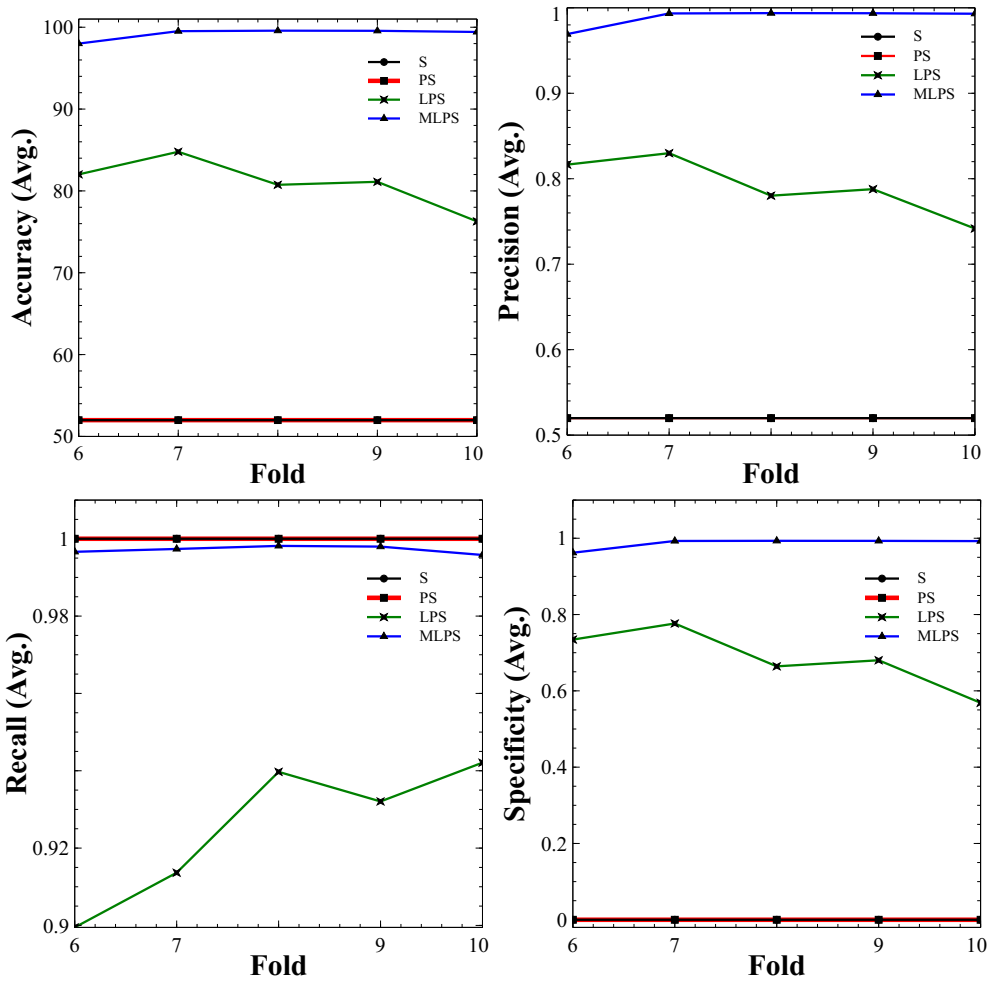


Figure 6.7: Classifier evaluation using the models in Figure 6.5 (AlgoViz Spring 2010 DSN).

We also check the precision of the classifier (see Equation 6.13) which is shown in Figure 6.6 (top right). Here again the *MLPS* model (the blue line) exhibits the highest precision (around 100%) compared to the other models. *LPS* (the green line) shows better precision compared to *S* and *PS*. Both *S* and *PS* models have approximately the same precision which is 60%. Both accuracy and precision for all of the models remain fairly similar for different folds. This indicates that the amount of data used to train and build a model has small impact from 6 to 10-folds.

Figure 6.6 (bottom left) shows the recall for the classifier. Recall is computed based on Equation 6.14. All of the models except *LPS* show a recall of 100% and overlap with each other. Only *LPS* shows a varying recall which ranges from 87% to 98%.

Lastly, Figure 6.6 (bottom right) shows the specificity of the classifier. Specificity shows the fraction of negative prediction for true negative cases (see Equation 6.15). In this plot, the *S* and *PS* models overlap at zero. The model with *LPS* generates values between 30% to 50%, showing an average performance. The best performance is achieved by the *MLPS* model which shows a specificity of

100%.

We also conducted these experiments with AlgoViz Spring 2010 data as seen in Figure 6.7. The accuracy for this dataset is given in Figure 6.7 (top left). Models S and PS both show an average accuracy of 52% for all the folds. The average accuracy of model LPS ranges from 74% to 86%. The highest average accuracy of LPS is 86% which is achieved at 7-fold and the lowest average accuracy is 76% at 10-fold. Model $MLPS$ gives the best accuracy of all the models, which is around 99% for any of the folds.

Figure 6.7 (top right) depicts the precision of the classifier. This plot shows a trend similar to the accuracy plot. When we start with only text similarity in the model the average accuracy is 52%. This increases as we add l, p, and m into the model. Model LPS shows an average precision between 72% to 84% while model $MLPS$ shows an average precision close to 100%.

Figure 6.7 (bottom left) shows the recall of the classifier. According to this measure, models S and PS achieve the highest recall which is close to 100%. Next best recall is achieved by model $MLPS$ which is around 99%. Model LPS exhibits the worst performance among all the models by producing a minimum recall of 90% at 6-fold and maximum recall of 94% at 10-fold.

Figure 6.7 (bottom right) shows the specificity of the models. Here the $MLPS$ model outperforms the other models. Models S and PS show a specificity of zero and overlap each other. Model LPS exhibits a specificity between 60% to 70% at various folds, showing a better performance than S and PS.

As we see from these two datasets, for the majority of cases, the classifier that has the best performance is the $MLPS$ model. The $MLPS$ model based classifier has an accuracy of 99% to 100% on average. The model also has around 99% precision in most cases. The average recall and specificity of this model is close to 100%. Compared to $MLPS$, the performance of LPS is low for all four of the evaluation measures. Models S and PS show similar performance, indicating that the inclusion of URLs in the model may not improve performance significantly.

One interesting point to note here is that although in most cases models S and PS perform poorly, they do exhibit very high recall, close to 100%. This is because these models produce an outcome of one for most cases. While training and testing the classifier we attempted to use similar numbers of positive and negative examples. Thus the models rightly predict the true outcomes in half the cases. Since recall does not consider false positives (see Equation 6.14) these models show a better performance according to the recall measure. They do however show a poor performance in specificity — a measure that considers the false positives (see Equation 6.15).

We also test the performance of the classifiers using the F1 score, which depends on precision and recall [112]. The value of F1 score ranges from 1 to 0 where 1 shows the best performance and 0 indicates worst performance. F1 score is computed as:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{6.16}$$

Figure 6.8 shows the F1 score for the Fall 2009 and Spring 2010 AlgoViz DSNs. The X axis shows the number of folds while the Y axis lists the average F1 score for that fold. In the Fall 2009 (Figure

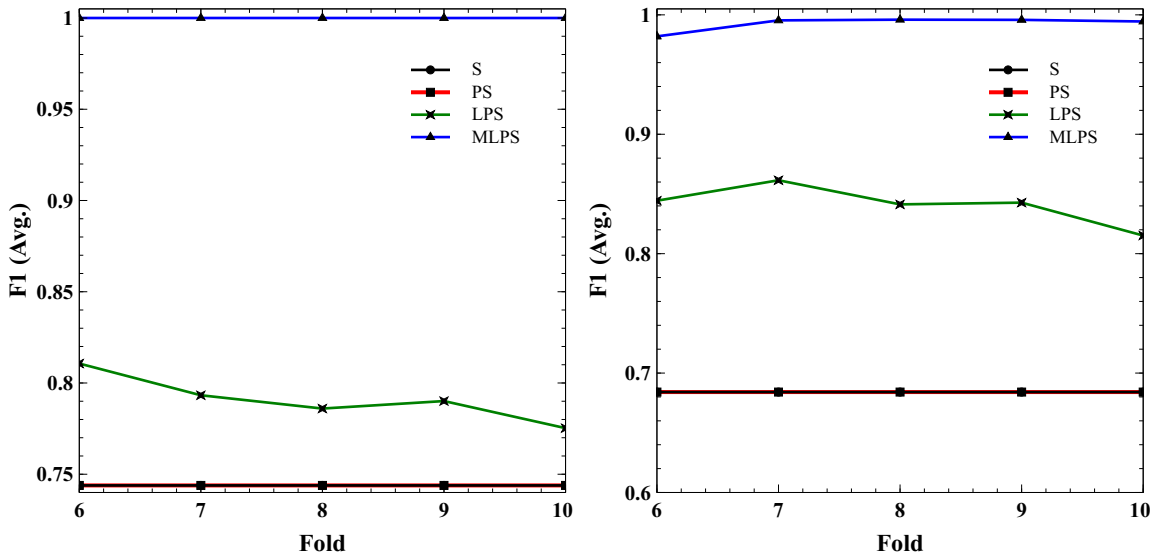


Figure 6.8: Average F1 score: (left) Fall 2009 DSN. (right) Spring 2010 DSN.

6.8 (left)) the lowest F1 score is achieved by the *S* and *PS* models which is 0.74. The *LPS* model shows improved F1 score (close to 0.8). The highest F1 score, which is 1, is achieved by the *MLPS* model. Similar trends are visible in the Spring 2010 DSN. Models *S* and *PS* exhibit similar F1 scores (approximately 0.74) while the *LPS* model shows improved performance — ranging from 0.82 to 0.85. The best F1 score is generated by the *MLPS* model. At 6-fold the F1 score for the *MLPS* model is 0.98 which increases to 0.99 for the subsequent folds. For all the DSNs, models *S* and *PS* have the worst F1 score and *MLPS* shows the best score. *LPS* performs in between these three models. Also, all models except *LPS* show a steady F1 score through all the folds. With higher number of folds, *LPS* shows decreasing F1 score. This indicates that even with more data the *LPS* fails to model user behavior successfully.

The ROC (Receiver Operating Characteristics) curves are also commonly used to evaluate classifier accuracy, especially for binary classifiers [30]. An ROC curve compares the true positive rates of a classifier with the false positive rates. An ROC curve shows *1-Specificity* vs. *Sensitivity*. The best model will have a value close to the point (0,1) — indicating high sensitivity and high specificity. Random guesses will result in points along the diagonal line from (0,0) to (1,1). A good classifier will result in points above the diagonal line.

Figure 6.9 shows the ROC curves for the AlgoViz Fall 2009 DSN and Spring 2010 DSN. True positives (i.e., recall) are computed using Equation 6.14 and true negatives (i.e., specificity) are computed using Equation 6.15.

The ROC curves for the 2009 AlgoViz DSN (Figure 6.9(left)) show that both the *S* and *PS* models falls across the diagonal line. *LPS* (green star) resides above the line indicating better accuracy. The best accuracy is achieved by the *MLPS* model (blue triangle). We see that the classifier performs poorly when we only consider text similarity (*S, PS*). The accuracy of the classifier increases when we include DSN-derived information (*L*). However, the best accuracy is achieved when DSNs are

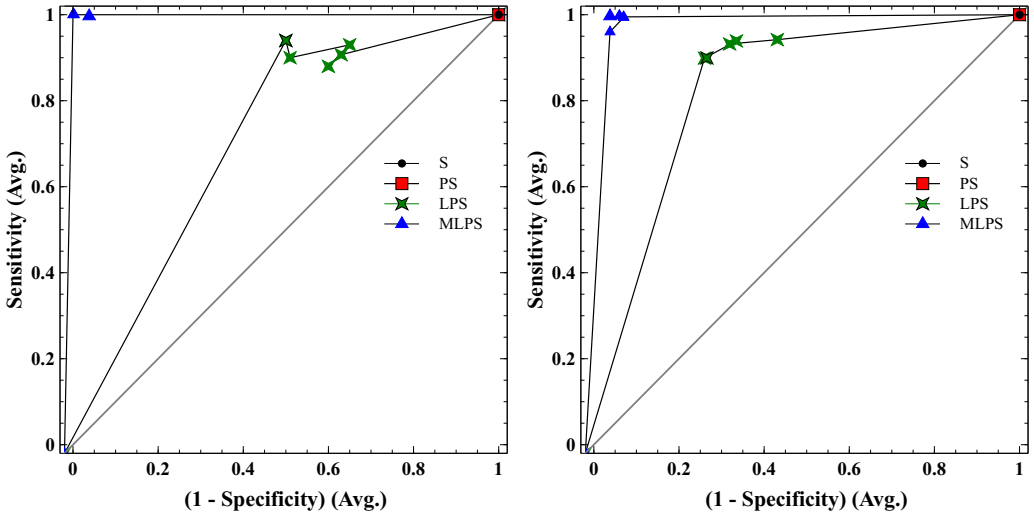


Figure 6.9: ROC curves: (left) Fall 2009 DSN. (right) Spring 2010 DSN.

broken down into groups and the group information is included in the model ($MLPS$). Figure 6.9(right) shows the ROC curves for the AlgoViz Spring 2010 DSN. The results are somewhat similar to that of Fall 2009 DSN. Here again, models S and PS fall along the diagonal line thus showing a poor performance. Compared to these two models LPS shows better performance by staying above the diagonal line in all the folds. The best performance is achieved by the $MLPS$ model which is closer to the (0,1) point compared to the other three models. One point to note here is that the distance between LPS and $MLPS$ is shorter compared to Fall 2009 DSN. This indicates that the Spring 2010 DSN provides a more informative DSN that aids in building a better model for the classifier.

Figure 6.10 shows the average runtime for building the classifiers using different models at different folds. For example, at 10 fold, it took 6 minutes to generate the classifiers using models $MLPS$, LPS, and S, while the classifier that used model PS took 7 minutes to build. On average, the classifiers built using various models generated from the AlgoViz Spring 2010 DSN took 6 minutes

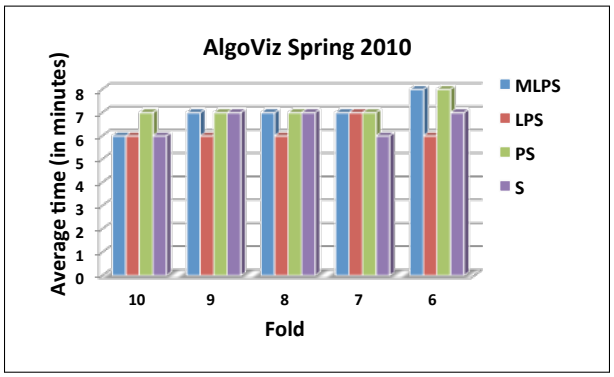


Figure 6.10: Average runtime for building classifiers using AlgoViz Spring 2009 DSN.

to build. However, the average runtime starts to increase at 6 fold where it takes 8 minutes to generate the classifiers with models $MLPS$ and PS, 7 minutes for the classifier with model S, and 6 minutes for the classifier with model LPS.

6.3.3 Recommendation performance

The classifiers described in the previous subsection are later used to recommend content. We take a similar approach in using n-fold cross validation to test the recommendation where we start at 10-fold and stop at 6-fold. The approach towards recommending resources based on DSN is described in Figure 6.3. We follow a similar approach while evaluating the recommendation framework.

For any given user u_i in the test fold, we have a list of pages that the user saw in a session. Based on the pages viewed by the user we assign him to a group. While building the DSN a number of users were not assigned to any groups for various reasons (e.g., did not view k similar pages like any other user, did not view k pages in a session). At this step we assign these users, who appear in the test fold, to a particular group based on their pageviews. The centroid method is used at this step; details can be found in Section 6.2.

Once a user is assigned to a group, we select one page ps_i from his session $ps_1, ps_2, \ldots, ps_n$ and one page pt_j from the set of pages used in the training phase, to build a *test page pair*, (ps_i, pt_j). This *test page pair* then passes through the resource pair proximity model which provides the probability of these two pages being viewed in a session. The estimated score is used to rank all the pairs. All the page pairs containing only the pages that appear in the session of user u_i are called *user page pairs*. We then compute the percentile of the *user page pairs* in the ranked list. The average percentile for all the *user page pairs* is reported for each fold. Our claim is that a good recommendation would place the *user page pairs* at the top of the ranking, thus in a higher percentile.

Figure 6.11 (left) shows the performance of the recommendation for the AlgoViz Fall 2009 dataset.

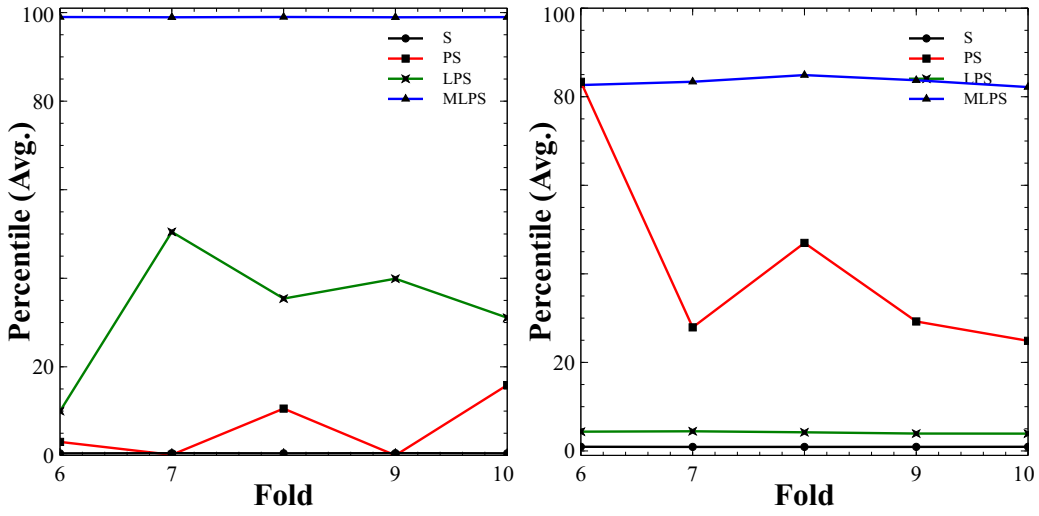


Figure 6.11: Recommendation evaluation: (left) Fall 2009 DSN. (right) Spring 2010 DSN.

When we use only similarity between the page titles to build the model, the recommendation framework does not perform well. The average percentile for *user page pairs* with model S is zero for all the folds. A slightly better but varying performance is seen by the next model, *PS*. According to this model, the average percentile for *user page pairs* is either zero or close to zero at 6-fold, 7-fold, and 9-fold. Folds eight and ten place those page pairs in around the 10th percentile. This means that the page pairs seen by the user appear at the bottom of the ranked list in Figure 6.3. Model *LPS* shows a varying but better performance than *PS*. At 6-fold it performs poorly where the *user page pair* appears in the 10th percentile. At 7-fold the *user page pairs* appear in the 35th percentile. For the rest of the folds the percentiles remain close to 25 for this model. With this dataset, the best performance is achieved using the *MLPS* which consistently places the *user page pairs* in the 99th percentile.

A similar trend is seen in the AlgoViz Spring 2010 dataset in Figure 6.11 (right). Model S performs poorly by placing the *user page pairs* low in the ranked list for all the folds. Model *PS* improves the recommendation performance slightly, which places the *user page pairs* in the 5th percentile. The performance of model *LPS* peaks at 6-fold with *user page pairs* appearing in the 80th percentile. However, as the number of folds increases the model starts to show decreasing performance. Model *MLPS* shows a steady and improved performance by placing the *user page pairs* at around the 80th percentile for each fold.

For both datasets, model *MLPS* performed better than the other models. While this model achieved a good recommendation performance for the Fall 2009 dataset (around 99th percentile) it showed a decreased performance for the Spring 2010 dataset (around 80th percentile). However, in both cases, it outperforms the other three models. One of the reasons for the decreased performance could be the large number of smaller groups in the Spring 2010 DSN. As we see in Figure 6.1, Spring 2010 has 12 groups of users. Many of these groups have two to six users. Having large numbers of groups with such few users increases the risk of inaccurate group assignment, eventually leading to recommendation of non-relevant content. One way to mitigate this problem is to merge smaller groups until they meet a certain user threshold. Another way could be to depend on the topic of the groups and to merge the groups as long as the topic distance is smaller than a certain distance.

6.4 Discussion and future work

Recommender systems depend on various user attributes to recommend resources. Such attributes may come from user profiles by parsing the demographic information or the preferences [72]. Explicit user activities such as ratings, reviews, or comments also act as features for building models to be used for recommendation [106]. Many recommendation systems suffer from the *Cold start problem* which refers to the lack of user feedback on resources. In cases where user feedback is available, it is difficult to recommend resources to new users who have not yet provided any feedback.

Lack of user attributes leads to the use of navigational information in recommendation systems. Chen [23] converts access patterns to two measures, Web access graph (WAG) and page interest estimator (PIE), to predict a user's interest in a certain page. Networks built using navigational information are used in many areas to model and predict user behavior [40, 39, 69]. However, many

of these networks depend on extensive user activities, such as building a graph of pages the user viewed [23], to build the model. Others depend on the page attributes, such as link structure [39], to provide recommendation. Contrary to these cases, our approach works with smaller sessions and does not depend heavily on the page attributes. As a page attribute we only rely on page title and URL. Using our approach, sessions with as little as two pageviews are sufficient to build the $MLPS$ model that shows promising performance in recommending resources.

Personalization in a digital library can take many forms. Personalization can be done based on individual user profiles, group membership, resource category, or perceived outcome [124]. Often digital libraries contain metadata for resources. Retrieving useful information from metadata can be difficult since the quality of metadata varies from data provider to data provider. In cases where the resources are present in the library, algorithms have been proposed to extract metadata that can be used in recommendation systems [59]. Although our proposed system is intended for digital libraries, it does not depend on any library-specific information, thus making it easily adaptable to other domains. Since we do not rely on profiles of registered users, our approach is particularly suitable for DLs with mostly anonymous users.

Our approach depends on the resource pair proximity model built using logistic regression. This model incorporates the group information derived from the DSNs. The approach is similar to the mixture model approach that is used in many domains including topic modeling [89], information retrieval [128], and trend prediction [139]. A mixture model is built using a linear combination of several different models. Although we use different predictor variables in the resource pair proximity model, we do not include different distributions into the model. Following the mixture model, one possibility to extend the resource pair proximity model might be to bring the distributions of the predictor variables into the model.

6.5 Summary

In this chapter we showed how the DSN can be used to build a model to recommend content. Our model depends on anonymous user activities to recommend content. This is the core area where our approach differs from traditional recommendation systems. We proposed an equation for building a DSN-based recommendation and used logistic regression to train and build the model. We performed various evaluations to test the performance of the model, the classifier that uses the model, and lastly the recommendation framework that utilizes the classifier. We also used different parameters to build different models and tested their performance. Results showed that using DSN-dependent parameters to build a recommendation system can improve the performance compared to when only text similarity is used. We believe that our approach has potential for educational DLs where implicit user activities (e.g., view, click, search) are abundant but explicit user activities (e.g., account creation, rating, comment) are low.

Chapter 7

Case Study: Ensemble - The Computing Portal

In the previous chapters we described the process of building DSNs from AlgoViz log data. We provided a series of analyses along with two applications that use the knowledge derived from the DSNs. Various evaluations showed promising performance when DSN-derived knowledge is used to enhance DL services. In this chapter, we provide similar analyses with another educational DL named Ensemble. We use Ensemble log data to generate DSNs following the same community detection framework described in Figure 4.2. Information on the detected communities is used to tailor Ensemble services. The following sections provide more detail on each step of the process, along with experimental results.

7.1 Ensemble: the computing portal

Ensemble[1] is a Pathways project of the National Science Digital Library (NSDL)[2] for computing education. Ensemble provides a distributed digital library for computing educators. It provides access to a broad range of computing educational resources, allows users to build or join communities, and hosts technologies that aid in teaching. Users can contribute resources; rate, tag, comment on, or share existing resources; and create or join communities. The growing collection of Ensemble resources includes ACM-W (ACM Women in Computing), AlgoViz, bjc (The Beauty and Joy of Computing), Nifty, BPC Engineering, CITIDEL, Computing History Museum, CSERD, CSTA, Digital library curricula, PAWS, PlanetMath, StemRobotics, SWNET, VKB, Walden's paths, and YouTube Education. Ensemble allows users to create communities. Most communities are open for joining, while some require membership approved by a group administrator. Groups can have their own forum, resources of different types (e.g., book posts, syllabi). Ensemble also hosts a section called *Technologies* that contains tools built as teaching aids.

[1] `https://computingportal.org/`

[2] `https://nsdl.org/`

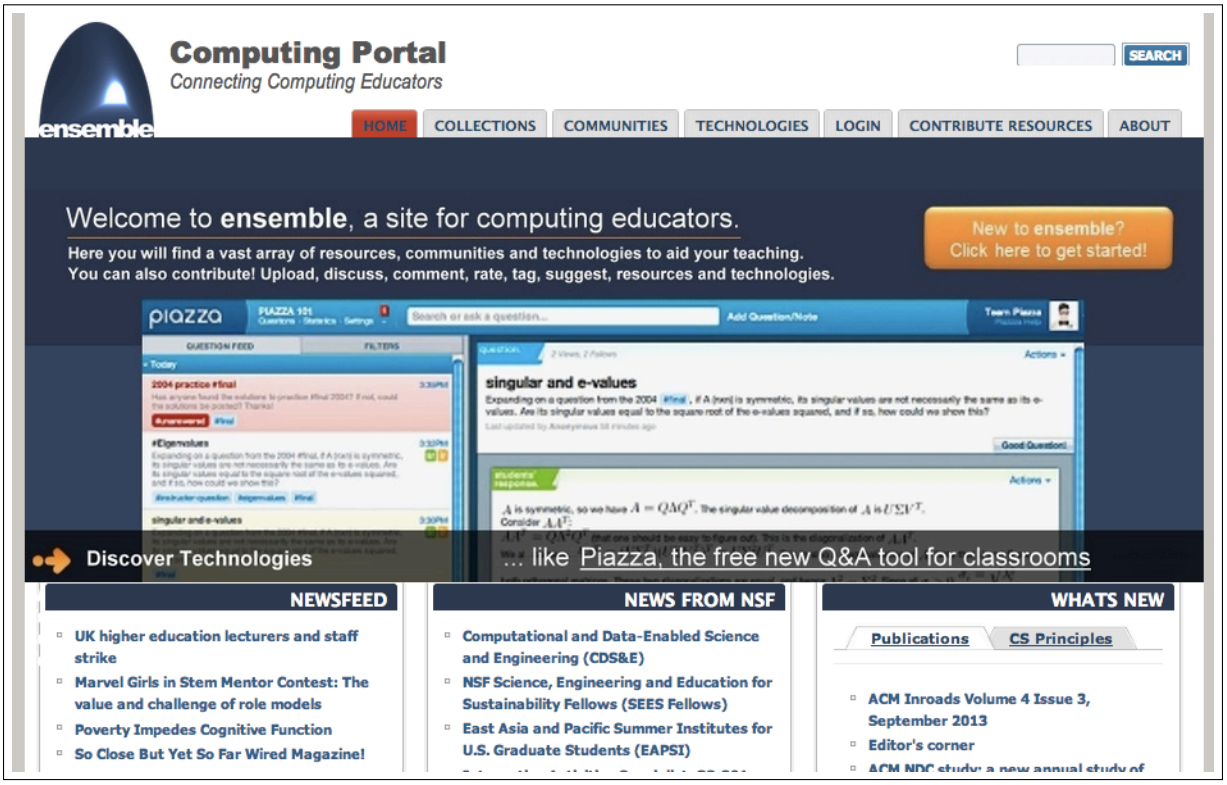


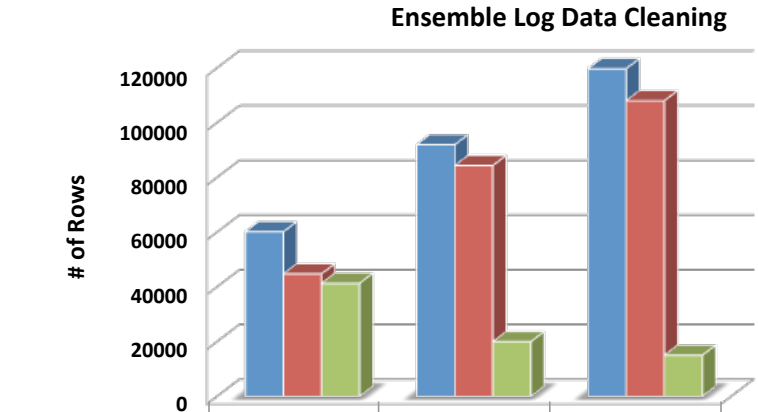
Figure 7.1: Ensemble front page.

Ensemble is implemented using the Drupal content management system[3]. Because Drupal is highly customizable, allowing developers to build new services or change existing ones, it is widely used for developing user-friendly systems. However, the default Drupal search in Drupal 6 provides less options to customize the service. Hence Ensemble uses the Solr search framework to provide faceted search to its users. Among many services Ensemble users are able to browse and search resources, create an account, contribute resources, subscribe to content to get notifications, and join or create groups. Various user activities are stored by Drupal modules in a number of tables (e.g., Accesslog, History, Watchdog). We selected to use the *accesslog* table for log analyses. Details of this table can be found in Section 4.3.1 of Chapter 4. In November 2013, Ensemble upgraded from Drupal 6 to Drupal 7 which provides faceted search options. The experiments in this chapter are done on the Ensemble data while it was in Drupal 6.

7.2 Network generation

We use log data from December 2011, February 2012, and August 2013 to build Ensemble DSNs. The *filtering module* cleans the log data using a three-step data cleaning process described in detail in Section 4.3.1.

[3] `https://drupal.org/`



	Aug 13	Feb 12	Dec 11
# rows after step 1	60181	92017	119772
# rows after step 2	44804	84403	108058
# rows after step 3	41365	20047	15071

Figure 7.2: Number of rows during data cleaning (see Section 4.3.1) of Ensemble DSNs.

Figure 7.2 shows the number of rows at various steps of the data cleaning process. The figure shows that the number of rows after the first stage of data cleaning in August 2013 is 60,181, in February 2012 is 92,017, and in December 2011 is 119,772. The second stage of data cleaning prunes 10,000 to 20,000 rows. The last stage of data cleaning is session-based filtering. As the chart shows, session-based filtering reduces around 3,000 rows in August 2013. However, the reduction in the number of rows is greater for the other two DSNs. Around 60,000 rows in February 2012 and 90,000 rows in December 2011 are pruned through session-based filtering. At the end, although the December 2011 DSN started with the highest number of rows, it ended with the lowest number of filtered rows (15,071) among the three DSNs. Of the three DSNs, the August 2013 one lost the least number of rows in the cleaning process indicating this month's log was less noisy — that is least affected by bots, crawlers, spammers, etc.

After the logs pass though the filtering module, we use the cleaned log data to generate the DSNs. We compute the network density (see Equation 4.1) at various connection thresholds for the three DSNs. Figure 7.3 shows how the density changes with varying connection threshold. The three lines represent the different DSNs. For all three DSNs we see a sharp drop in density when the connection threshold is increased from two to four. The drop is most noticeable for the August 2013 DSN. Also, the density of the DSNs reach close to zero after $k = 4$. We opted to use a k between two and four. At $k = 2$ the DSN may contain more edges, many of which will be less informative since users will be connected if they viewed two common pages. Hence we choose to use $k = 4$ for further analyses.

Figure 7.4 shows network attributes such as nodes and edges of the DSNs with connection threshold $k = 4$. Among the three DSNs, the August 2013 has the most nodes (users) at 4,218, and the December 2011 has the least nodes at 1,637 (see column 1 in Figure 7.4). The August 2013 DSN also has the highest number of edges — 11,622 — which is close to the number for the December 2011 DSN (see column 2 in Figure 7.4). The number of users connected via an edge (*Connected*

Nodes) in December 2011 is 396, in February 2012 is 260, and in August 2013 is 466 (see column 3 in Figure 7.4). The number of isolated nodes, i.e., nodes without any edge, is much higher than the number of connected nodes (see column 4 in Figure 7.4).

Figure 7.5 shows the DSNs generated using $k = 4$. The left DSN is for December 2011 and the middle network represents the February 2012 DSN. The DSN at right is for August 2013. The plots show only the connected nodes and omit isolated nodes. The color and size of the nodes are proportional to their degree (i.e., number of edges connected to a node). The December 2011 DSN (Figure 7.5 (left)) shows the presence of two distinct groups with a large number of edges between them. The February 2012 DSN (Figure 7.5 (middle)) also exhibits two groups, however the sizes of the groups are close to each other, whereas in the December 2011 DSN the bottom group was larger compared to the top group. The August 2013 DSN (Figure 7.5 (right) shows one large central group with numerous smaller groups. This indicates that while there is a large group of users with shared interest, there also are smaller groups of users with different interests.

7.3 Network partitioning

Figure 7.5 gave us a glimpse of the groups of users in each DSN. In this section we use graph partitioning algorithm to define those groups. We use LinLog layout [95, 96], as we did with the AlgoViz DSNs, which uses modularity to partition the network. Figure 7.6 (top-left) shows the number of users in each cluster found in the Ensemble December 2011 DSN. As we see in this plot, the first two clusters have more than 100 users. The next two clusters have close to 60 users. Each of the remaining clusters has less than ten users. This indicates that among the groups detected in the network, only a few have a large number of users. Compared to these large groups, Ensemble shows the presence of more smaller groups. Even though the target audience for Ensemble is computing educators, the diversity in their navigational history shows that there is a large number of small

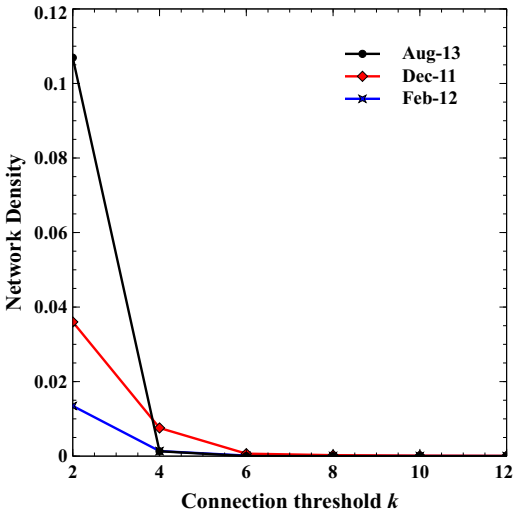
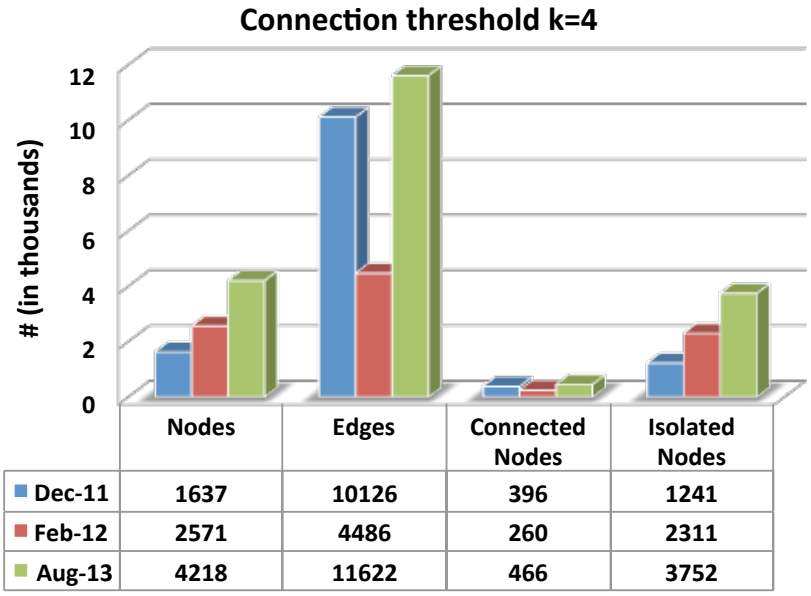


Figure 7.3: Network density for varying connection threshold in Ensemble.



	Nodes	Edges	Connected Nodes	Isolated Nodes
Dec-11	1637	10126	396	1241
Feb-12	2571	4486	260	2311
Aug-13	4218	11622	466	3752

Figure 7.4: Network attributes of Ensemble DSNs.

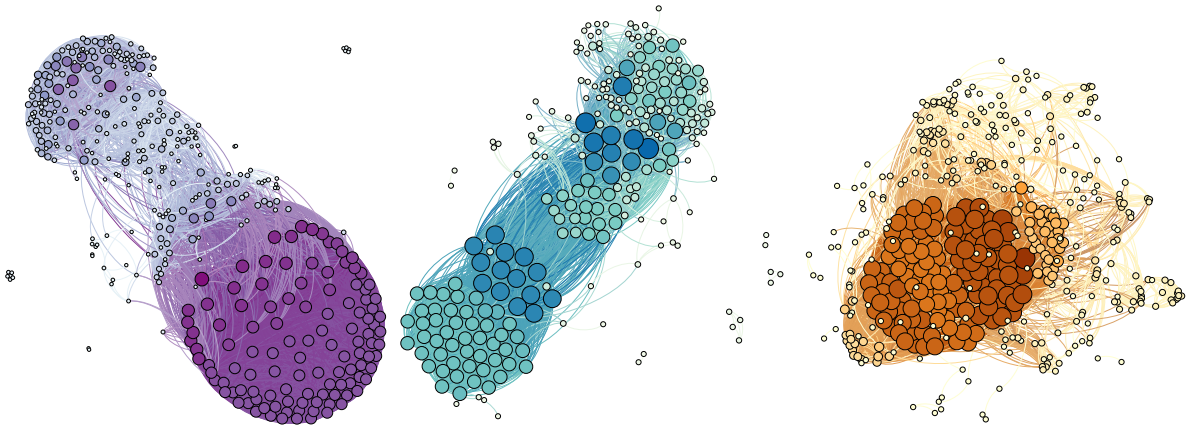


Figure 7.5: Ensemble DSNs: (left) December 2011, (middle) February 2012, (right) August 2013.

groups of educators with distinctive navigational trends.

Similar to December 2011, we found ten clusters in February 2012 DSN. The user distribution in those clusters is shown in Figure 7.6 (top-right). Among the ten clusters detected, only three have 40 or more users (i.e., Clusters #1, #2, and #3). The rest of the clusters contain between two and five users. Both the December 2011 and February 2012 DSN show a trend of three to four large user groups containing more than 40 users and six to seven smaller user groups containing less than ten users. We see a slightly different trend with the August 2013 DSN clusters portrayed in Figure 7.6 (bottom). The partitioning of this DSN resulted in 27 groups. Out of these 27 groups

there are four groups with more than 40 users (i.e., Clusters #2, #3, #5, and #6). There are two groups with more than 20 users (i.e., Clusters #4 and #7). The rest of the groups have less than 20 users. Thus, similar to the other DSNs in this figure we see around six groups with more than 20 users coexisting with a larger number of smaller groups. Again we see the presence of many smaller groups along with a few big groups.

7.4 Revised ranking

In Chapter 5 we showed the potential of using DSN-based revised ranking with search results. In this section we provide two similar case studies in Ensemble to show how our approach of ranking search results using DSN-derived information performs with the Solr search ranking. To test these two rankings we created the benchmark rank called *log-based rank*. Log-based rank uses click counts of external links within a page. Our assumption is that a user will click the outgoing link only when he finds the resource potentially interesting or useful. Here clicks act as a positive feedback from the user on the potential usefulness of the page viewed.

While the revised ranking uses pageviews to detect user behavior, we also created a pageview-only rank and compared it with the log-based rank. For each page returned by the Solr search, we computed its pageview from the timespan of the DSN we use - December 2011. One point to note with the pageview-based rank is that various activities including searching, browsing, following a

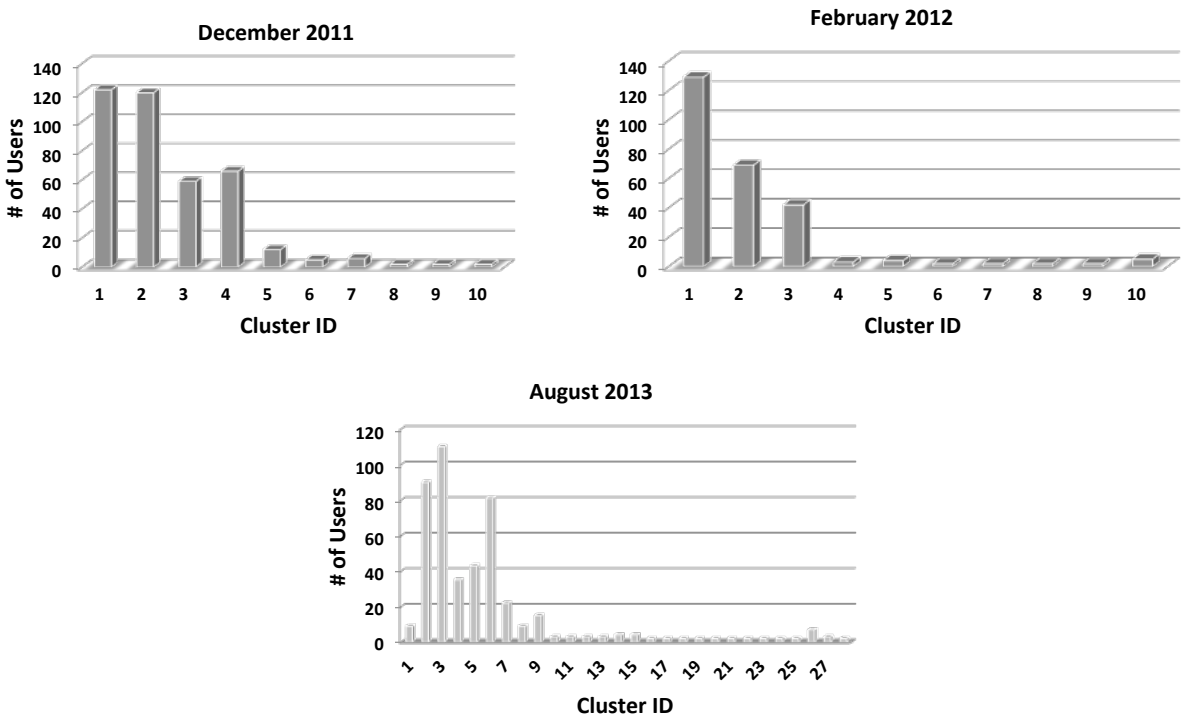


Figure 7.6: # of users in groups: (top-left) Dec-11, (top-right) Feb-12, (bottom) Aug-13.

link from, email, or a post on the Web, etc. can generate pageviews. Thus, although a pageview shows user interest in a page, it is difficult to detect how much of any activity, such as search, contributed to the total pageviews of a page.

Compared to the two search systems in AlgoViz, Ensemble relies on one — the Solr search engine. The results of Solr are ordered using two particular attributes of a page — recency of the page and the number of comments in the page. A page that is created recently will be promoted over a page which was posted earlier. Also, pages with more comments are favored over pages with less to no comments. Lastly, pages with more recent comments also are given more weight.

In an effort to identify ranking performance we tested the Solr search ranking, revised ranking, and pageview-based ranking with the log-based rank. The revised ranking uses the DSN from December 2011. December 2011 is one of the DSNs with fewer groups. Since the DSNs we are using have one month's data, the amount of information in the DSN is already low compared to AlgoViz where we used a couple of months of data. Less data and more groups have the potential to provide lesser information on smaller groups. Hence, we opted to use a DSN with a low number of groups (i.e., December 2011 has 10 groups). The pageviews are computed within the same DSN (i.e., December 2011).

We used two types of coefficients, social interest coefficient and weighted interest coefficient, introduced in Chapter 5, to generate the revised ranking. The social interest coefficient, s, provides information on the variety of the user groups that viewed a page. On the other hand the weighted interest coefficient, p, within a group indicates how this page compares to the most-viewed page of this group. These coefficients are computed as:

$$s = \frac{\text{\# of groups containing the resource } re_i}{\text{\# of total groups}} \tag{7.1}$$

$$p = \frac{\text{\# of users who viewed resource } re_i \text{ in group } g_j}{\text{\# of users who viewed the most viewed resource(s) in group } g_j} \tag{7.2}$$

Any given resource will have one social interest coefficient (s) and multiple weighted interest coefficients (p) — one for each group in the DSN. These coefficients are used in the following equation that computes the *DSN score* for each page of a search result:

$$\text{DSN score}_{re_i} = s_{re_i} \times \sum \{p_{re_1}, p_{re_2}, \ldots, p_{re_n}\} \times 100 \tag{7.3}$$

where n is the number of groups in the DSN. The DSN scores are used to rank the pages. Note that Equation 7.3 of DSN score is the same as the DSN-based ranking in Chapter 5.

We selected two search terms that appeared in the search log in Ensemble: *Merge sort* and *Floyd*. An Ensemble search returns a list of ranked results for these query terms. *Merge sort* returned 35

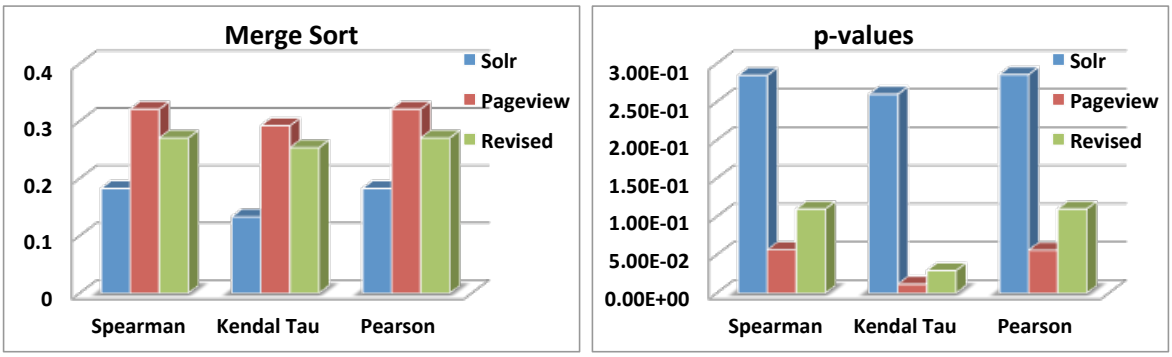


Figure 7.7: Revised ranking using search term *Merge Sort* for the December 2011 DSN.

pages while *Floyd* returned five pages. For each of these queries we computed the log-based rank, pageview-based rank, and revised rank. All three rankings (Solr, pageview-based rank, and revised rank) were compared against the log-based rank using three rank evaluation measures (Spearman's rho, Kendall's tau, and Pearson's correlation coefficient). We provide more details of these measures in Section 5.3.

Figure 7.7 shows the results of the analyses for the search term *Merge sort*. The left plot shows the values for the three evaluation measures for each rank when compared to the log-based rank. Each bar points to one of the three ranks that is compared with the log-based rank. The p-values for each evaluation are shown in the plot on the right. As we see in the left plot, the Solr rank has the lowest Spearman's rho value among the three ranks. Clearly, the revised rank performs better than the Solr rank. However, the best rho values are achieved by the pageview-based rank.

The significance of the ρ values can be determined using the p-values in the plot on the right. This plot shows that each of the three ρ values has a p-value that is equal to or higher than 0.05. Thus, according to Spearman's rho, none of the ranks show any significant correlation with the log-based rank.

The middle set of bars in the left plot portrays the Kendall's tau value for each of the three ranks. Here again, Solr has the lowest τ value while pageview-based rank shows the highest τ value. The

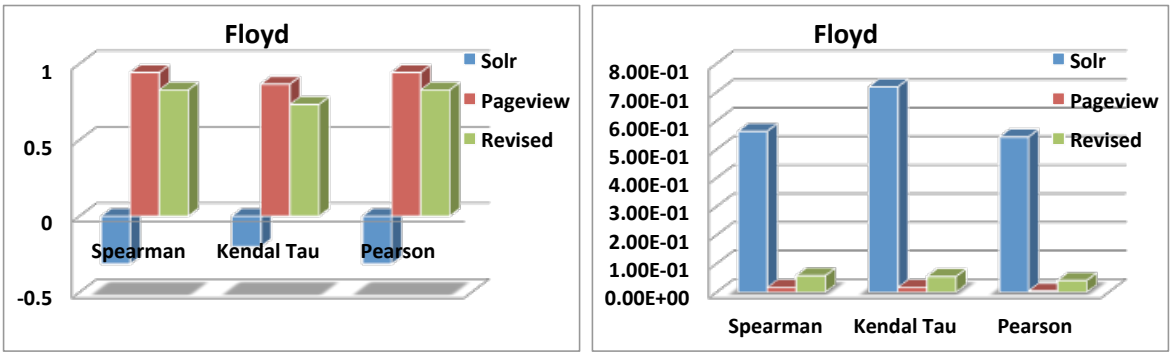


Figure 7.8: Revised ranking using search term *Floyd* for the December 2011 DSN.

τ value of revised rank is closer to the pageview-based rank. When we look at the corresponding p-values on the plot to the right, we see that both the pageview-based rank and revised rank have p-values lower than 0.05, but the Solr rank shows much higher p-values. Thus, Kendall's tau indicates both revised rank and pageview-based rank have strong correlation with the log-based rank.

The last set of bars, at the right side of the left plot, shows the Pearson's correlation coefficient values for the three ranks. The coefficients and their p-values (plot on the right) are identical to the Spearman's rho value.

The results of the second case study with the search term *Floyd* is shown in Figure 7.8. The Solr rank shows negative correlation with the log-based rank in all three evaluation measures. The associated p-values are higher than 0.05 which fail to indicate significant correlation between Solr rank and log-based rank. The pageview-based rank produces the highest coefficients among all three measures (see Figure 7.8(left)). The corresponding p-values are less than 0.05 (see Figure 7.8(right)) indicating significant correlation between pageview-based rank and the benchmark log-based rank. Lastly, the revised rank produces coefficients in the range of 0.7 to 0.8 in the three evaluation scale. The p-values of these coefficients range from 0.04 to 0.05 indicating significant correlation.

These cases indicate that similar to AlgoViz, the Solr rank system used by Ensemble fails to show significant correlation with the log-based rank. Thus there is scope for improvement in the search result ranking. Contrary to AlgoViz, Ensemble does not have any custom ranking. In AlgoViz the best ranking was achieved by combining the custom score with the DSN-derived information. In Ensemble, both the revised rank and the pageview-based rank demonstrate better performance than the default systems. However, as we said earlier, the pageviews can be biased by a range of activities. DSNs filter these activities by considering sessions with similar pageviews. Also, the revised ranks show significant correlation with the log-based rank. From this preliminary study, we see that DSN-based revised rank is less effected by noisy data, better captures user interest, and thus has strong potential to improve existing ranking of edu-DLs.

7.5 DSN-based recommendation

In this section we use the three DSNs from December 2011, February 2012, and August 2013, to build a model for the recommender system. In Chapter 6 we used four models with different predictor variables. However, in this chapter, we use the complete model with all the predictor variables, *MLPS*, and the model with one variable related to similarity between page titles, S. We compare how these two models perform for different DSNs. Note that the DSNs contain isolated users who are not connected. The session information of the isolated users of the DSNs is included in the $MLPS$ model through the variable l. The following sections describe experimental results of the model evaluation along with the performance of the corresponding classifiers, and of the recommender system.

7.5.1 Evaluation of the model

Our aim for the recommender system that uses the information gained from a DSN is for a user u_i who belongs to group g_k of the DSN to find the likelihood of viewing page p_n, given that he viewed page p_i. The model we use for this section was described earlier in Chapter 6. It can be stated as:

$$\begin{aligned} & c_1 \times \text{Similarity between titles } (p_i, p_n) + \\ & c_2 \times \text{Longest common prefix } (LCP) \text{ in URLs } (p_i, p_n) + \\ & c_3 \times l(p_i, p_n) + \\ & c_4 \times m(p_i, p_n, g_k) \end{aligned} \tag{7.4}$$

where $c_1, c_2, \ldots, c_4$ are the coefficients, $l(p_i, p_n)$ is the number of times pages (p_i, p_n) appear in all sessions in the DSN, and

$$m(p_i, p_n, g_k) = \frac{\#\text{ of times } (p_i, p_n) \text{ appears in sessions in } g_k}{\#\text{ of users in } g_k}. \tag{7.5}$$

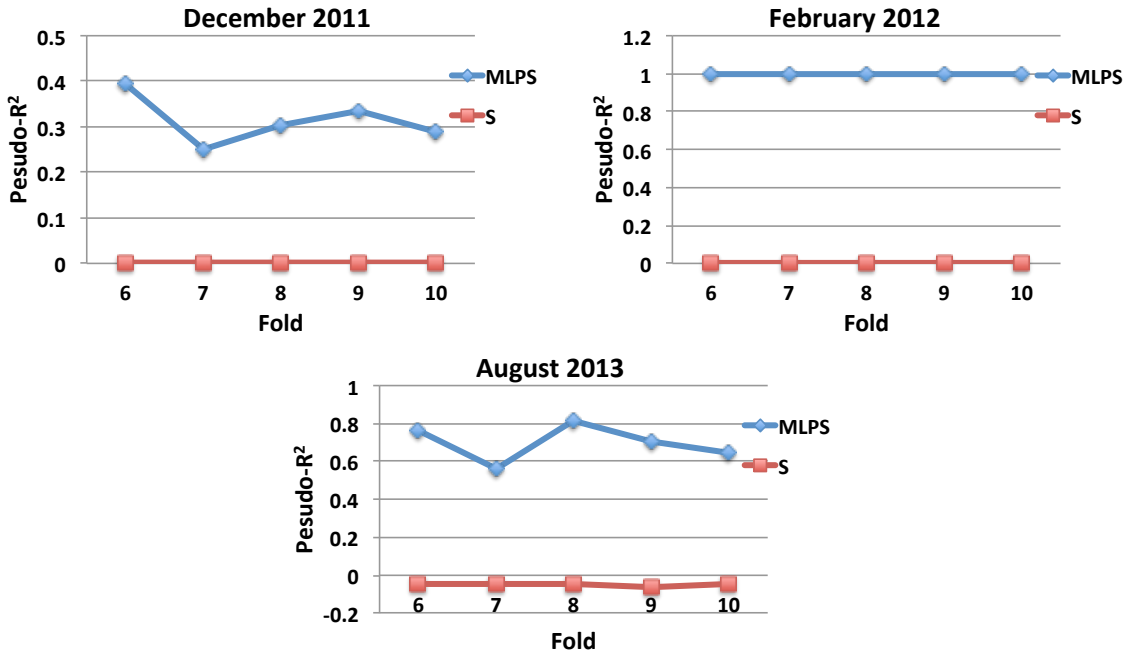


Figure 7.9: Pseudo-R^2 for the *MLPS* and *S* models of three DSNs.

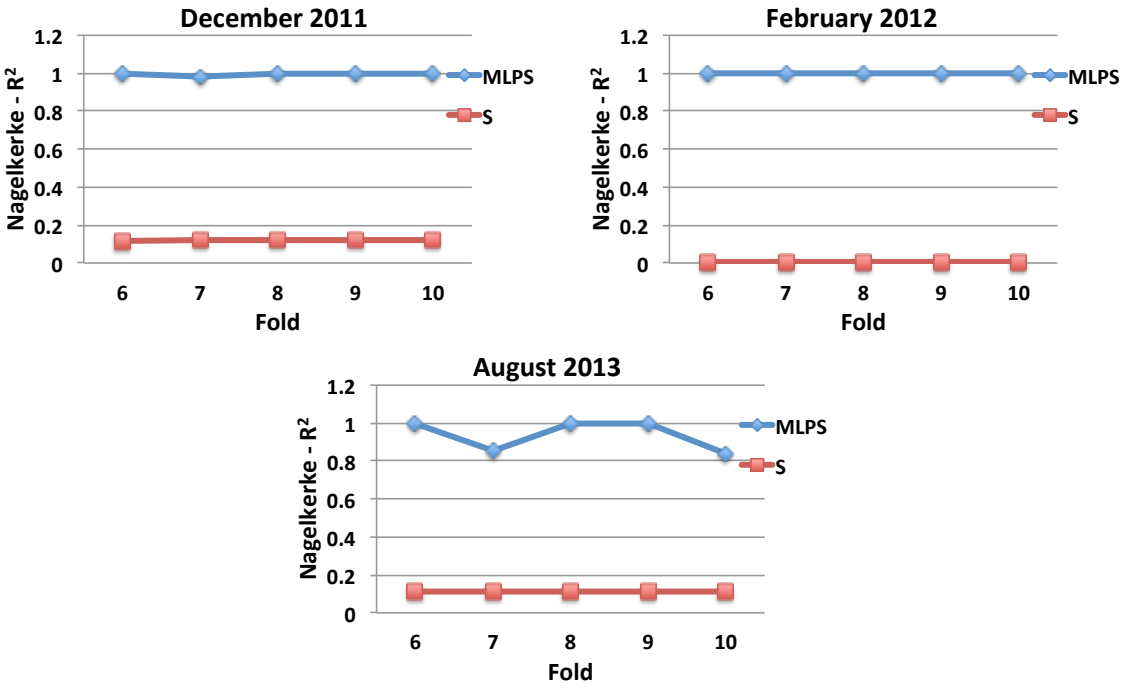


Figure 7.10: Nagelkerke-R^2 for the *MLPS* and S models of three DSNs.

Table 7.1: Coefficients of the resource pair proximity model for December 2011 DSN at 10 fold.

Model	Intercept	M	L	P	S
MLPS	6.98	850.62	-0.17	-0.13	3.69
LPS	-6.39		1.81	-2.34	3.38
PS	-2.18			-1.65	3.67
S	-2.97				1.60

The similarity between a pair of pages is measured using Cosine similarity [42]. The values of l and m are computed from the corresponding DSN. With these predictor variables we use logistic regression to build the model, which we refer to as the *resource pair proximity model*. As an example, Table 7.1 shows the coefficients of different models at 10 fold for the December 2011 DSN.

We used two approaches, Pseudo-R^2 [76] and NagelKerke R^2 [92], to test the goodness of fit for the models. Compared to Chapter 6, in this chapter we skipped the Cox and Snell R^2 measure since NagelKerke is an adjusted form of Cox and Snell. Details of these measures are described in Section 6.3.1.

Figure 7.9 shows the Pseudo-R^2 values for the resource pair proximity models for the three DSNs. The top-left plot shows the results obtained using the *MLPS* and S models for the December 2011 DSN. The top-right plot shows similar results over February 2012 DSN. The bottom plot shows results from the August 2013 DSN. As we see, in December 2011, the *MLPS* model performs better than the only text similarity-based model (S). The Pseudo-R^2 value is highest for the *MLPS* model at 6 fold and lowest at 7 fold. After 7 fold the Pseudo-R^2 value increases with each fold. The model

built with the August 2013 DSN also shows a similar trend, where the Pseudo-R^2 value drops at 7 fold. The S model for this DSN performs poorly compared to the *MLPS* model and produces Pseudo-R^2 values that are close to zero for each of the folds. The Pseudo-R^2 value of the *MLPS* model at 6 fold is 0.8 which decreases to 0.6 at 7 fold. The behavior of the model using February 2012 data is a little different. The *MLPS* model using this DSN produces a steady Pseudo-R^2 value that is close to one.

The same *MLPS* model using the same DSNs show a slightly different trend when we use Nagelkerke to evaluate the goodness of fit of the model. The results of this evaluation are shown in Figure 7.10. According to this figure, the *MLPS* model results in steady Nagelkerke-R^2 value closer to one for both the December 2011 and February 2012 DSN. The S model for December 2011 is close to 0.1 for each fold. However, the S model for February 2012 is closer to zero. This indicates that when used alone, the text-based model may not always contribute to a good model. A similar trend is visible in the August 2013 DSN where Nagelkerke-R^2 values for the *MLPS* model range from 0.8 to 1 for the folds and the S model is close to 0.1.

These results indicate that when compared to the S model, the *MLPS* model provides a better fit. For some DSNs, the behavior of this model changes slightly with the amount of data used but the overall R^2 value remains similar over different folds. Contrary to this behavior, the performance of the S model is steady across all folds for the various DSNs we tested. This model often produces R^2 values closer to zero indicating that the amount of data used in training this model has no effect on the quality of the model.

7.5.2 Evaluation of the classifier

We used the models of the previous section to test their performance as a classifier. At each step of the evaluation, we used n-1 folds for building the model and one fold for testing the performance of the model to successfully identify if two pages will appear together in a session. We used four evaluation measures to test the performance: precision, recall, F1 score, and accuracy. Precision shows the percentage of correct predictions for the positive examples. Recall provides the fraction of positive cases that are predicted as positive. Precision and recall are used to compute the F1 score for the classifiers. Lastly, we compute the accuracy of the classifiers. While precision and recall both provide different information on the performance of the classifier for the positive data and positive predictions, accuracy provides an overall indication of the classifier performance by providing the fraction of accurate predictions for both positive and negative data. Details of each of these measures are provided in Section 6.3.2.

The precision of the three models is described in Figure 7.11. The X-axis of the plots shows the folds while the Y-axis shows the average precision for each fold. The average precision is measured on a scale of 0 to 1 where 1 indicates a classifier with best precision performance. The top-left plot shows the average precision of the models using the December 2011 DSN. The S model has an average precision close to 0.6 for all the folds. Compared to the S model, $MLPS$ performs better as it achieves an average precision close to one for all the folds. This model however does show a slight decrease in average precision at 7 fold.

We see a similar trend with the February 2012 and August 2013 DSNs. The average precision of

the S model for February DSN is closer to 0.5 which is slightly lower than the other DSNs. Unlike the $MLPS$ model with the December DSN, the $MLPS$ models in the other two DSNs do not show any visible decrease in precision at any fold.

While the solely text-based S model did not perform as well as the $MLPS$ model in terms of precision, we see a different trend when we use *recall* to evaluate the models. Figure 7.12 shows the recall of the models using different DSNs. The folds are plotted on the X-axis and the average recall values are plotted on the Y-axis. As we see, in all three DSNs, both the S and $MLPS$ models achieve the highest recall values (hence the lines overlap). Recall shows how a classifier performs in finding the true positives whereas precision shows how precisely the classifier performs in finding those true positives. While the good recall of the S model is encouraging, the lower precision of this model makes it less reliable. The $MLPS$ model on the other hand shows the ideal tendency of high precision and high recall values, making this model more reliable than the S model.

Figure 7.13 shows the average F1 score of the three DSNs. F1 score uses both precision and recall. A good classifier will maximize both precision and recall and produce a favorable score according to the F1 measure. A classifier that performs moderately on both precision and recall will have a good F1 score compared to a classifier that performs extremely well in one of these measures (precision, recall) and shows poor performance in another. As we see in Figure 7.13, $MLPS$ achieves an F1 score of one for all the DSNs. However, the F1 score of S in December 2011 is 0.71, in February 2012 is 0.67, and in August 2013 is 0.76. All the plots show that $MLPS$ performs better than the S model.

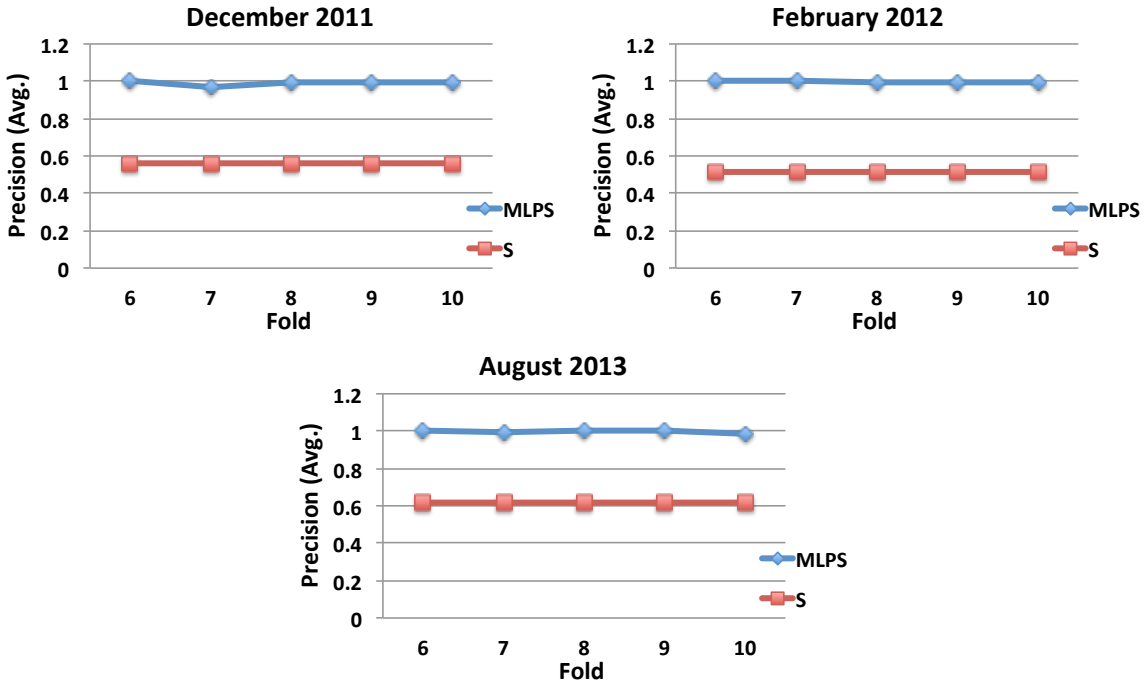


Figure 7.11: Precision of the classifiers using the $MLPS$ and S models of three DSNs.

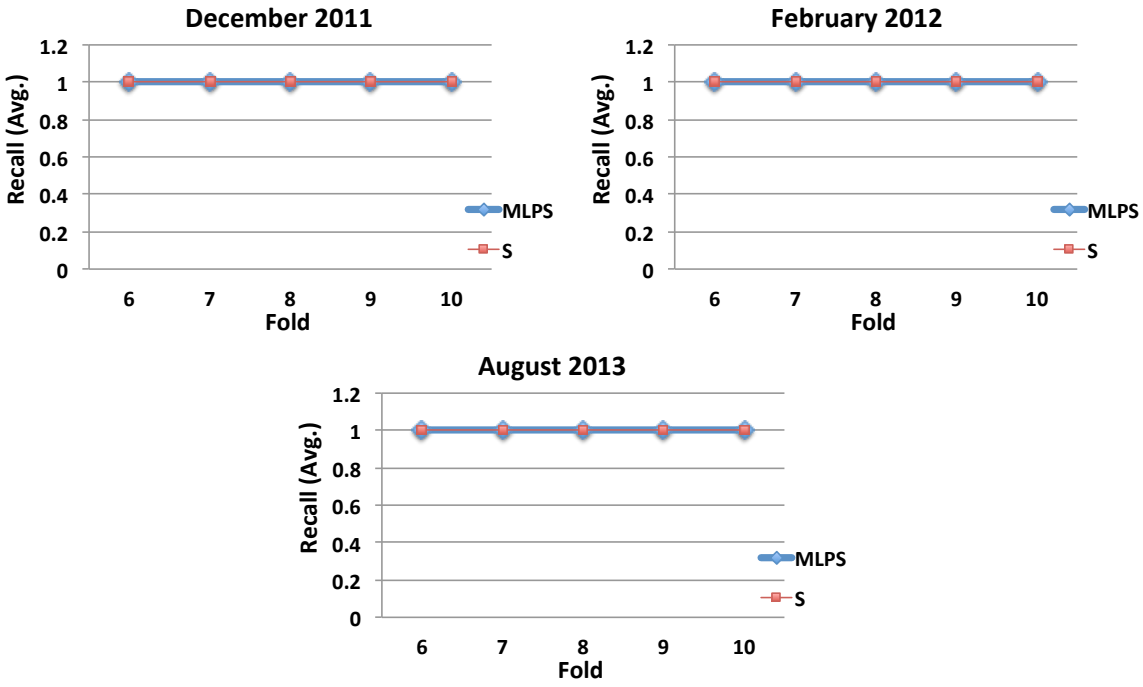


Figure 7.12: Recall of the classifiers using the *MLPS* and *S* models of three DSNs.

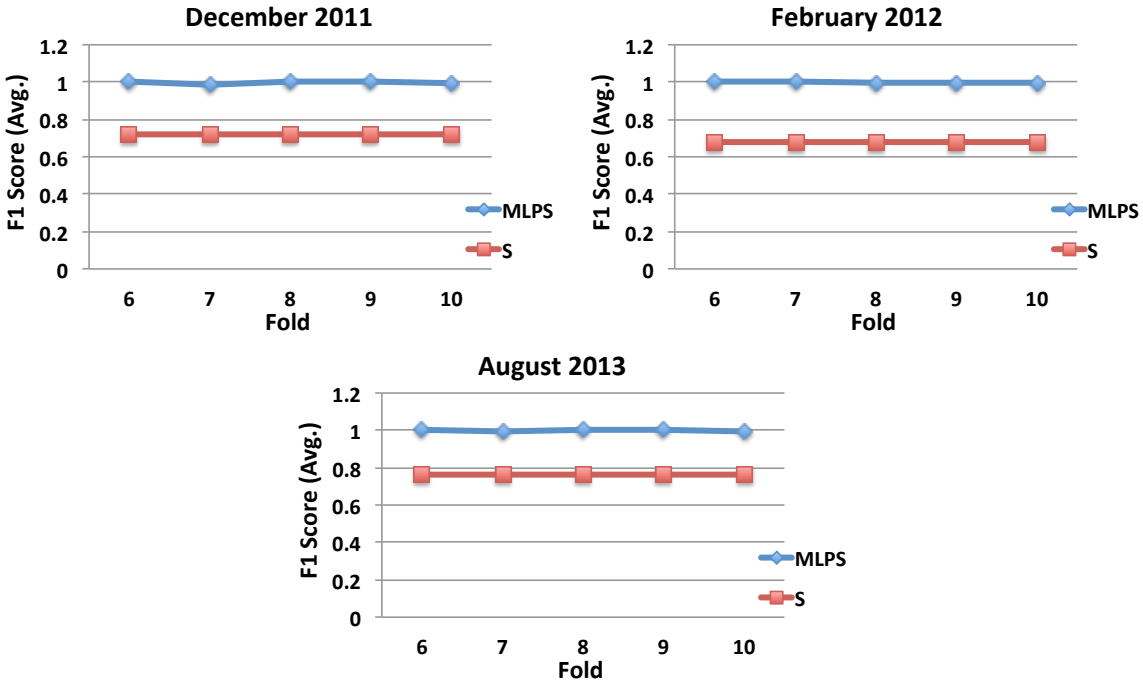


Figure 7.13: F1 score of the classifiers using the *MLPS* and *S* models of three DSNs.

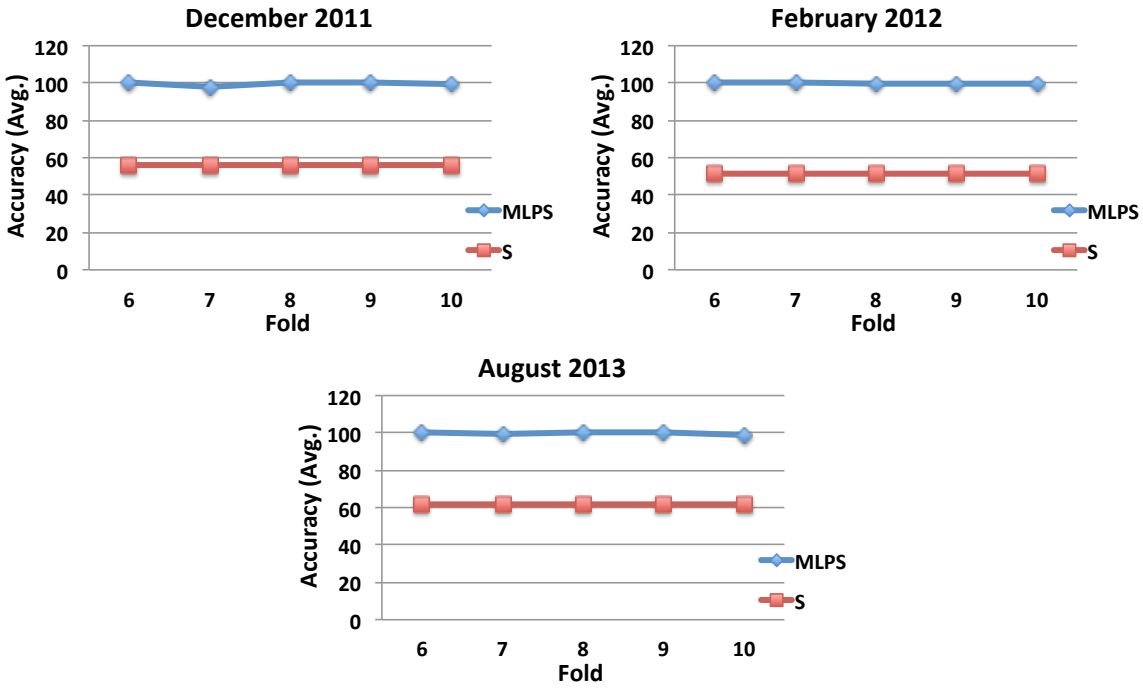


Figure 7.14: Accuracy of the classifiers using the *MLPS* and *S* models of three DSNs.

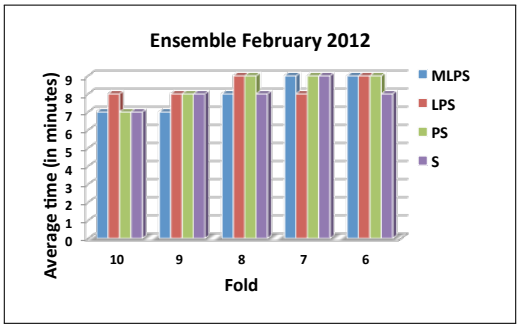


Figure 7.15: Average runtime for building classifiers using Ensemble February 2012 DSN.

Lastly, Figure 7.14 shows the accuracy of the classifier for the two models using different DSNs. Accuracy shows the percentage of correct predictions both positive and negative. The plots show that the accuracy of the *MLPS* model is close to 100%. However, the accuracy of the model *S* varies depending on the DSN and remains at or below 60% for the DSNs we used.

Based on various evaluation measures, we can conclude that for the three Ensemble DSNs, the *MLPS* model performs better than the *S* model. Also, from 6 to 10 folds the amount of training data does not significantly affect the performance of the models.

The average runtime for building the classifiers based on Ensemble data is shown in Figure 7.15. At 10 fold, it takes around 7 minutes to build the classifiers using the Ensemble February 2012 DSN. At lower folds, the average runtime increases from 7 minutes to 9 minutes.

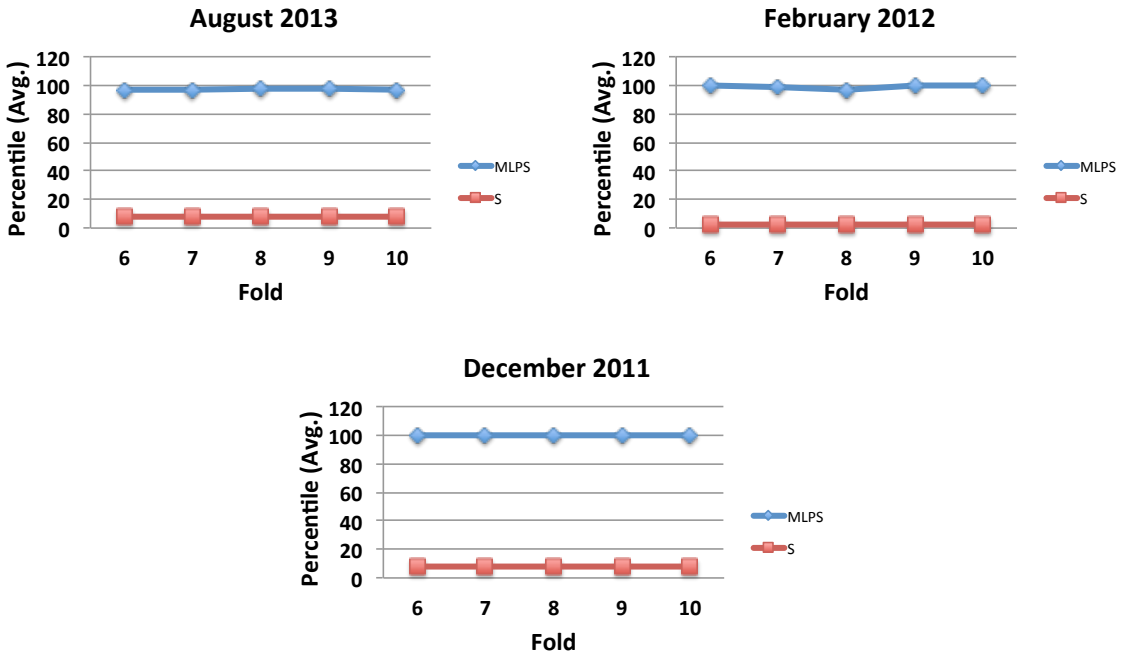


Figure 7.16: Evaluating the Recommender for the three DSNs.

7.5.3 Evaluation of the recommendations

We used the same models that we discussed in the previous section to test their performance as a recommender using different DSNs. To evaluate the performance of the recommender system, we used the testing fold and computed the percentile of the truly viewed pages among all the predicted pages. For each user in the testing fold, if he was not placed in a group by the partitioning algorithm (that is, he is an isolated user), we assigned him to an existing group (see Section 6.2). We then used his session information to build page pairs. For example, suppose a user in the testing fold viewed pages p_3 and p_8 together in a session. For each page, p_3 and p_8, using the $MLPS$ model, we computed the likelihood of a set of page-pairs, such as $(p_3, p_x), (p_3, p_y) \ldots$, where p_x, p_y, appear in the training data, appearing together in a session. This provides us with a ranked list of page-pairs. Among all the page-pairs we compute the percentile of page-pair (p_3, p_8), actually seen by the user, in the ranked list. The page-pair (p_3, p_8) is generated twice during the computation, once for page p_3 and once for page p_8. The average percentile for page-pair (p_3, p_8) is then reported in the plots. Our claim is that a good recommender system will place a page-pair that is visited by a user, such as (p_3, p_8), in the higher percentile.

Figure 7.16 shows the average percentile of the page pairs found using the $MLPS$ and S models for different DSNs. As we see, in all the plots, $MLPS$ is able to place the true pairs among the top percentile. The performance of this model does not vary from 6 to 10 fold. However, the S model places the true pairs below the 20th percentile in most cases and close to the 0th percentile in the February 2012 DSN. Overall, across different DSNs and different amounts of training data (from 6 folds to 10 folds), the performance of the recommender system using the $MLPS$ model is

significantly better compared to the performance of the text-based S model.

7.6 Summary

In this chapter we used log data from Ensemble to generate three DSNs. We used month-long log data selected from three different months of three years: December 2011, February 2012, and August 2013. We described various steps of building the DSNs and later described two possible applications that can incorporate DSN-derived information. One application, DSN-based revised ranking, showed promising performance compared to the existing ranking scheme. With the other application, recommender system, we showed that the model that uses DSN-derived information performs better than the model that relies only on text information.

Chapter 8

Conclusion

In this chapter we summarize the dissertation by presenting the problems that we addressed and our contributions. We conclude this chapter by describing future research directions.

Educational digital libraries are being developed in the hope of assisting educators, students, researchers, developers, policy makers, and other groups of people to create, use, reuse, and disseminate educational resources. With advances in technology, the information needs of the users of edu-DLs also are changing. Educational DLs that intend to effectively serve users need to keep up to date with the technology and demands of its users. Increasingly large amounts of educational resources make it difficult for users, particularly educators, to locate and use quality educational material. Peer review is an important way of identifying quality educational resources. However, due to the lack of an active educator community that reviews, provides feedback, or rates the resources, edu-DLs often do not have enough peer-reviewed materials. In the absence of explicit user feedback, we need to rely more on implicit usage data to deduce user interest.

8.1 Contributions

In this research we studied and identified key aspects of the next generation of edu-DLs. We formally define these DLs, present an approach called deduced social network (DSN) for modeling one of the key areas of these DL, and show the feasibility of using DSNs in edu-DL through two applications. We present an approach to incorporate DSN-derived information with the existing ranking system within an edu-DL. We also present a DSN-based recommendation application that relies on anonymous user logs to recommend content. Both applications show enhanced performance when DSN-derived information is incorporated. Next we briefly describe each of our contributions.

1. In order to identify current resource-seeking trends of educators and their information needs we conducted focus groups. The data from the focus groups helped us identify the short-comings of current edu-DLs and key aspects of the next generation of edu-DLs.
2. Based on our findings we proposed a formal definition for the next generation of edu-DLs that we call *edu-DL 2.0*. To be useful and effective, these DLs need to connect users and resources

in different ways to promote different types of interactions.

3. One important aspect of edu-DL 2.0 is that it should foster online communities. We proposed a formal definition of online community. In light of edu-DL 2.0 and the user interactions in them, we present a rubric to evaluate online communities within an edu-DL.

4. We investigated four edu-DLs and the communities within them. We described how they perform with regard to the formal definitions of edu-DL 2.0 and online community.

5. We proposed a graph representation of the interaction between users and the resources within an educational DL, denoted as a deduced social network (DSN). These graphs have the potential to reveal useful information on user behavior and trends. They can be particularly useful for edu-DLs where user feedback is important but often missing.

6. We showed that DSN-derived information can improve the existing search result ranking in AlgoViz. The DSNs we used were generated from pageviews. However, they provide more information on the viewing trends of a resource. With the help of a DSN, we can not only find out if different groups of users viewed a resource, but we also can detect the level of user interest in that resource in different groups.

7. We also proposed a DSN-dependent content recommendation framework for edu-DLs. While there exist many recommender systems, what makes our approach different is that we solely rely on anonymous user data (hence limited user features) and our target audience (i.e., educators within an educational DL) is fairly small and less diverse. Both of these factors make it difficult to effectively model users. However, we show that DSNs can be successfully used to recommend content.

8.2 Future work

This research has potential to be expanded in a number of directions. We briefly describe some areas for further exploration.

1. User activities can be explicit (e.g., comments, ratings) or implicit (pageviews, downloads). In this research we used implicit user activities and used a particular set of attributes (user, pageview, and time) from the log data to build the DSNs. An edu-DL rich in other types of user activities can use different sets of attributes (comments, downloads) to create different types of DSNs which might reveal different trends and usage patterns.

2. Bi-clustering or co-clustering techniques [86] simultaneously cluster the same dataset from two dimensions. Similar to this approach, multiple DSNs of different types can be analyzed together to reveal potentially interesting information. For example, along with the user-user DSN we used in this dissertation we also can create page-page DSNs and investigate these two DSNs together.

3. DSNs use a connection threshold to vary the strength of the network. We tested network characteristics (e.g., components, edges) with varying connection thresholds. Further research can

be done on automatically selecting the optimum connection threshold depending on network characteristics.

4. The implicit user activities of a large or popular edu-DL can rise exponentially with time. Depending on the timespan of the DSN and the size of the user-base, scalability of DSNs is another area that can be explored further.

5. We selected two services, ranking and recommendation, to investigate the feasibility of using DSNs in improving the performance of those services. Further research on how DSNs can effect other services can be beneficial in improving other DL services.

6. Our proposed DSN framework depends on archived log data. Further research can be done on implementing the DSN framework in existing DLs, where the DSNs dynamically change.

We believe adequate dissemination of information on resource usage trends can help the users of edu-DLs in locating useful resources. If the best or common practices are not explicitly documented, analysis of user trends can aid us to deduce common practices. In the absence of adequate user feedback, our proposed approach can help edu-DLs in finding common usage patterns. We believe our approach can be successfully deployed in other educational portals to discover hidden trends and tailor services accordingly. We hope that scalable implementations of search and recommender solutions that leverage our findings can be incorporated in the AlgoViz and Ensemble systems, as well as other digital libraries.

Bibliography

[1] Serge Abiteboul, Mihai Preda, and Gregory Cobena. Adaptive on-line page importance computation. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 280–290, 2003.

[2] L. B. Adajian. Connecting research to teaching: Professional communities: Teachers supporting teachers. *Mathematics Teacher*, 89(4):321—324, 1996.

[3] Gediminas Adomavicius and YoungOk Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans. Knowl. Data Eng.*, 24(5):896–911, 2012.

[4] A. Agresti. *Categorical data analysis*. Wiley, New York, 2002.

[5] Dong Aijuan and Wang Baoying. Domain-based recommendation and retrieval of relevant materials in e-learning. In *IEEE International Workshop on Semantic Computing and Applications (IWSCA 2008), 10-11 July 2008*, pages 103–108, Los Alamitos, CA, USA, 2008.

[6] Monika Akbar, Weiguo Fan, Clifford A. Shaffer, Yinlin Chen, Lillian Cassel, Lois Delcambre, Daniel D. Garcia, Gregory W. Hislop, Frank Shipman, Richard Furuta, B. Stephen Carpenter, Haowei Hsieh, Bob Siegfried, and Edward A. Fox. Digital library 2.0 for educational resources. In *Proceedings of the 15th International Conference on Theory and Practice of Digital Libraries: Research and Advanced Technology for Digital Libraries*, pages 89–100, 2011.

[7] Monika Akbar, Clifford A. Shaffer, and Edward A. Fox. Deduced social networks for an educational digital library. In *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '12, pages 43–46, 2012.

[8] Alias-i, LingPipe 4.1.0. http://alias-i.com/lingpipe/, [last visited on December 15, 2013].

[9] D G Altman and J M Bland. Statistics notes: Diagnostic tests 1: sensitivity and specificity. *BMJ*, 308(6943):1552, 6 1994.

[10] D. C. Andrews. Audience-specific online community design. *Communications of the ACM*, 45(4):64–68, 2002.

[11] José Augusto Azevedo, Maria Emília O. Santos Costa, Joaquim João E.R. Silvestre Madeira, and Ernesto Q. Vieira Martins. An algorithm for the ranking of shortest paths. *European Journal of Operational Research*, 69(1):97 – 106, 1993.

[12] M. Balabanovi and Y. Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[13] G. Beenen, K. Ling, X. Wang, K. Chang, D. Frankowski, P. Resnick, and R. E. Kraut. Using social psychology to motivate contributions to online communities. In *Proceedings of the 2004 ACM conference on Computer Supported Cooperative Work*, pages 212–221, 2004.

[14] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[15] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of community hierarchies in large networks. *CoRR*, abs/0803.0476, 2008.

[16] C. L. Borgman, M. J. Bates, M. V. Bates, E. N. Efthimiadis, A. J. Gilliland-Swetland, Y. B. Kafai, G. H. Leazer, and A. B. Maddox. Social aspects of digital libraries. Final Report for Invitational NSF workshop held at UCLA, February 1996. `http://is.gseis.ucla.edu/research/dig_libraries/UCLA_DL_Report.html` [last visited on December 15, 2013].

[17] Christine L. Borgman. Social aspects of digital libraries (working session). In *Proceedings of the First ACM International Conference on Digital Libraries*, page 170, Bethesda, Maryland, USA, 1996.

[18] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, 1998.

[19] Brian S. Butler. Membership size, communication activity, and sustainability: A resource-based model of online social structures. *Information Systems Research*, 12(4):346–362, 2001.

[20] L. Candela, D. Castelli, N. Ferro, Y. Ioannidis, G. Koutrika, C. Meghini, P. Pagano, S. Ross, D. Soergel, M. Agosti, M. Dobreva, V. Katifori, and H. Schuldt. The DELOS Digital Library Reference Model - Foundations for Digital Libraries. Version 0.98, 2008. `http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf` [last visited September 24, 2013].

[21] Lillian (Boots) Cassel, Ed Fox, Frank Shipman, Peter Brusilovsky, Weiguo Fax, Dan Garcia, Greg Hislop, Richard Furuta, Lois Delcambre, and Sridhara Potluri. Ensemble: Enriching communities and collections to support education in computing: Poster session. *Journal of Computing Sciences in Colleges*, 25(6):224–226, June 2010.

[22] CCSDS. Reference Model for an Open Archival Information System (OAIS): Recommendation for Space Data System Standards: CCSDS 650.0-B-1. Technical report, Consultative Committee for Space Data Systems, January 2002.

[23] Philip K. Chan. Constructing Web user profiles: A non-invasive learning approach. In *Web Usage Analysis and User Profiling, International WEBKDD'99*, pages 39–55, 1999.

[24] D. N. Chen and Y. C. Chiang. A document recommendation system based on collaborative filtering and personal ontology. In *11th International Conference on Informatics and Semiotics in Organisations*, pages 255–262, April 2009.

[25] Jian Chen, Jian Yin, and Jin Huang. Automatic content-based recommendation in e-commerce. In Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, pages 748–53, Los Alamitos, CA, USA, 2005.

[26] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):1–6, 2004.

[27] Thomas F. Coleman and Jorge J. Moré. Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM Journal on Numerical Analysis*, 20(1):187–209, 1983.

[28] David Roxbee Cox and E. Joyce Snell. *Analysis of binary data*. Monographs on statistics and applied probability. Chapman & Hall, London, Weinheim, New York, 1989.

[29] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 519–528.

[30] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 233–240, 2006.

[31] C. S. de Souza and J. Preece. A framework for analyzing and understanding online communities. *Interacting with Computers*, 16(3):579–610, 2004.

[32] W. H DeLone and E. R. McLean. Information systems success: The quest for the dependent variable. *Information Systems Research*, 3(1):60–95, 1992.

[33] A. Dieberger, P. Dourish, K. Höök, P. Resnick, and A. Wexelblat. Social navigation: techniques for building more usable systems. *interactions*, 7(6):36–45, November 2000.

[34] P. Domingos and M. Pazzani. Beyond independence: conditions for the optimality of the simple Bayesian classifier. In *Proceedings of Thirteenth International Conference on Machine Learning (ICML), 3-6 July, 1996*, pages 105–112, San Francisco, CA, USA.

[35] DSpace. DSpace homepage, 2003. `http://www.dspace.org/` [last visited September 24, 2013].

[36] R. O. Duda and P. E. Hart. *Pattern classification and scene analysis*. Wiley-Interscience, Chichester, Sussex, UK, 1973.

[37] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 281–285, 1988.

[38] C. Eds. Lagoze, H. Van De Sompel, M. Nelson, and S. Warner. The Open Archives Initiative Protocol for Metadata Harvesting – Version 2.0. 2002. `http://www.openarchives.org/OAI/openarchivesprotocol.html` [last visited on December 15, 2013].

[39] I. Esslimani, A. Brun, and A. Boyer. From social networks to behavioral networks in recommender systems. In *International Conference on Advances in Social Network Analysis and Mining, (ASONAM '09)*, pages 143–148.

[40] Ilham Esslimani, Armelle Brun, and Anne Boyer. Densifying a behavioral recommender system by social networks link prediction methods. *Social Network Analysis and Mining*, 1(3):159–172, 2011.

[41] Edward A. Fox, Yinlin Chen, Monika Akbar, Clifford A. Shaffer, Stephen H. Edwards, Peter Brusilovsky, Dan Garcia, Lois Delcambre, Felicia Decker, David Archer, Richard Furuta, Frank Shipman, Stephen Carpenter, and Lillian Cassel. Ensemble PDP-8: Eight principles for distributed portals. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, JCDL '10, pages 341–344, 2010.

[42] William B. Frakes and Ricardo Baeza-Yates, editors. *Information Retrieval: Data Structure and Algorithms*. Prentice Hall, 1992.

[43] Jill Freyne, Rosta Farzan, and Maurice Coyle. Toward the exploitation of social access patterns for recommendation. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 179–182, 2007.

[44] Fuhr, N. and Hansen, P. and Mabe, M. and Micsik, A. and Sølvberg, I. Digital libraries: A generic classification and evaluation scheme. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pages 187–199, 2001.

[45] William Gale. Good-Turing smoothing without tears. *Journal of Quantitative Linguistics*, 2:217–237, 1995.

[46] A. Girgensohn and A. Lee. Making Web sites be places for social interaction. In *Proceedings of the 2002 ACM conference on Computer Supported Cooperative Work*, pages 136–145, 2002.

[47] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[48] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

[49] M. A. Gonçalves, E. A. Fox, L. T. Watson, and N. A. Kipp. Streams, structures, spaces, scenarios, societies (5S): A formal model for digital libraries. *ACM Transactions on Information Systems*, 22:270–312, April 2004.

[50] M. A. Gonçalves, B. L. Moreira, E. A. Fox, and L. T. Watson. "What is a good digital library?" - a quality model for digital libraries. *Information Processing and Management: an International Journal*, 43:1416–1437, September 2007.

[51] Marcos André Gonçalves and Edward A. Fox. 5SL: a language for declarative specification and generation of digital libraries. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, JCDL '02, pages 263–272, 2002.

[52] Daniel Greenstein and Suzanne Thorin. *The Digital Library: A Biography*. Digital Library Federation Council on Library and Information Resources, 2002. `http://www.clir.org/PUBS/reports/pub109/pub109.pdf` [last visited on December 20, 2013].

[53] Stanford DLI Group. Stanford University Digital Libraries Project. http://www-diglib.stanford.edu/diglib/ [last visited on December 15, 2013].

[54] David Gurzick and Wayne G. Lutters. Towards a design theory for online communities. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, pages 11:1–11:20, 2009.

[55] Tom Habing. DLI Publications. http://dli.grainger.uiuc.edu/pubsnatsynch.htm [last visited on December 15, 2013].

[56] Jeremy Hadidjojo and Siew Ann Cheong. Equal graph partitioning on estimated infection network as an effective epidemic mitigation measure. *PLoS ONE*, 6(7):e22124, 2011.

[57] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[58] Bruce Hendrickson and Robert Leland. An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM Journal on Scientific Computing*, 16(2):452–469, 1995.

[59] Francis Heylighen and Johan Bollen. Hebbian algorithms for a digital library recommendation system. In *Proceedings 2002 International Conference on Parallel Processing Workshops, IEEE Computer*, pages 72–75. Society Press, 2002.

[60] Z. Huang, W. Chung, T. H. Ong, and H. Chen. A graph-based recommender system for digital library. In *Joint Conference on Digital Libraries (JCDL)*, pages 65–73, Portland, Oregon, USA, 2002.

[61] T. K. Huwe. Exploiting synergies among digital repositories, special collections, and online community. *Online*, 33(2):14–19, 2009.

[62] Hidehiko Ino, Mineichi Kudo, and Atsuyoshi Nakamura. Partitioning of Web graphs by community topology. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 661–669, 2005.

[63] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323, September 1999.

[64] L. Barbato S. Clark Y. George S. Hsi C. Lowe P. Mackinney K. Lightle, E. Almasy and E. McIlvain. Draft report: Metrics recommendations and resources for NSDL projects, 2009. `http://nsdl.library.cornell.edu/websites/nsdlnetwork/sites/default/files/Draft%20Report-Metrics_Recommendations_0.pdf` [last visited on December 20, 2013].

[65] Shlomo Kalish and Paul Nelson. A comparison of ranking, rating and reservation price measurement in conjoint analysis. *Marketing Letters*, 2(4):327–335, 1991.

[66] M. G. Kendall. A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81–93, June 1938.

[67] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, September 1999.

[68] J. Koh, Y. Kim, B. Butler, and G. Bock. Encouraging participation in virtual communities. *Communications of the ACM*, 50:68–73, February 2007.

[69] Arnd Christian König, Michael Gamon, and Qiang Wu. Click-through prediction for news queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 347–354, 2009.

[70] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[71] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. On social networks and collaborative recommendation. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 195–202, 2009.

[72] Bruce Krulwich. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45, 1997.

[73] Shailendra Kumar, Sajal K. Das, and Rupak Biswas. Graph partitioning for parallel applications in heterogeneous grid environments. In *Parallel and Distributed Processing Symposium - (IPDPS'02)*, pages 66–72, 2002.

[74] C. Lagoze, D. Krafft, S. Payette, and S. Jesuroga. What is a digital library anymore, anyway? beyond search and access in the NSDL. *D-Lib Magazine*, 11(11):1082–9873, November 2005.

[75] C. Lagoze and H. V. Sompel. Compound information objects: the OAI-ORE perspective. *Open Archives Initiative Object Reuse and Exchange*, 2007. White Paper, `http://www.openarchives.org/ore/documents` [last visited on December 15, 2013].

[76] Thomas Laitila. A Pseudo-R^2 measure for limited and qualitative dependent variable models. *Journal of Econometrics*, 56(3):341 –355, 1993.

[77] J. M. Leimeister, P. Sidiras, and H. Krcmar. Success factors of virtual communities from the perspective of members and operators: An empirical study. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 37); January 5-8 2004; Hawaii*, volume 7, pages 2708–2715, 2004.

[78] G. Lekakos and P. Caravelas. A hybrid approach for movie recommendation. *Multimedia Tools Appl.*, 36(1-2):55–70, 2008.

[79] H. Lin. Determinants of successful virtual communities: Contributions from system characteristics and social factors. *Information and Management*, 45(8):522–527, 2008.

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

[81] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*, IUI '10, pages 31–40, 2010.

[82] Huang Longjun, Dai Liping, Wei Yuanwang, and Huang Minghe. A personalized recommendation system based on multi-agent. In *2008 Second International Conference on Genetic and Evolutionary Computing (WGEC), 25-26 Sept. 2008*, pages 223–226, Piscataway, NJ, USA, 2008.

[83] P. J. Ludford, D. Cosley, D. Frankowski, and L. Terveen. Think different: Increasing online community participation using uniqueness and group dissimilarity. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 631–638, 2004.

[84] M. Markland. Technology and people: Some challenges when integrating digital library systems into online learning environments. *The New Review of Information and Library Research*, 9(1):85–96, 2003.

[85] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7(3):216–244, July 1960.

[86] I Van Mechelen, H H Bock, and P De Boeck. Two-mode clustering methods: a structured overview. *Statistical methods in medical research*, 13(5):363–394, October 2004.

[87] Prem Melville, Raymod J. Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA, 2002.

[88] D. R. Millen and J. F. Patterson. Stimulating social engagement in a community network. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, pages 306–313, 2002.

[89] David R. H. Miller, Tim Leek, and Richard M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 214–221, 1999.

[90] Raymond J. Mooney and Loriene Roy. Content-based book recommending using learning for text categorization. *Proceedings of the ACM International Conference on Digital Libraries*, pages 195–204, 2000.

[91] A. Mufit Ferman, Peter Van Beek, James H. Errico, and M. Ibrahim Sezan. Multimedia content recommendation engine with automatic inference of user preferences. In *Proceedings of International Conference on Image Processing, ICIP-2003, September 14-17, 2003*, volume 3, pages 49–52, Barcelona, Spain, 2003.

[92] N. J. D. NagelKerke. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, 1991.

[93] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[94] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, February 2004.

[95] Andreas Noack. Energy models for graph clustering. *Journal of Graph Algortihms and Applications*, 11(2):453–480, 2007.

[96] Andreas Noack. Modularity clustering is force-directed layout. *CoRR*, abs/0807.4052, 2008.

[97] 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.

[98] Jeremy E. Oakley and Anthony O'Hagan. Probabilistic sensitivity analysis of complex models: A Bayesian approach. *Journal of the Royal Statistical Society, Series B*, 66:751–769, 2002.

[99] Mahsa Orang and Nematollaah Shiri. A probabilistic approach to correlation queries in uncertain time series data. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, pages 2229–2233, 2012.

[100] Tim O'Reilly. What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software, September 2005. `http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html` [last visited on December 15, 2013].

[101] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[102] F.C. Pampel. *Logistic Regression: A Primer*. SAGE Publications, 2000.

[103] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[104] Sandra Payette and Carl Lagoze. Flexible and Extensible Digital Object and Repository Architecture (FEDORA). In *ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 41–59, London, UK, 1998. Springer-Verlag.

[105] Sandra Payette and Thornton Staples. The Mellon Fedora Project. *Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries, Lecture Notes in Computer Science 2458, Springer*, pages 406–421, 2002.

[106] M. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408, 1999.

[107] M. J. Pazzani and D. Billsus. Content-based recommendation systems. In *The Adaptive Web. Methods and Strategies of Web Personalization*, pages 325–341. Springer, Berlin, Germany, 2007.

[108] David M. W. Powers. Evaluation: From precision, recall and f-factor to ROC, informedness, markedness & correlation. Technical Report SIE-07-001, School of Informatics and Engineering, Flinders University, Adelaide, Australia, 2007.

[109] J. Preece, B. Nonnecke, and D. Andrews. The top five reasons for lurking: Improving community experiences for everyone. *Computers in Human Behavior*, 20(2):201–223, 2004.

[110] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-Validation. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 532–538. Springer US, 2009.

[111] Joerg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74:016110, 2006.

[112] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.

[113] Stephen E. Robertson and Karen Sparck Jones. *Document retrieval systems*. Taylor Graham Publishing, London, UK, 1988. Chapter Relevance weighting of search terms, pages 143–160.

[114] V. Robles, P. Larranaga, E. Menasalvas, M. S. Perez, and V. Herves. Improvement of naive Bayes collaborative filtering using interval estimation. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), 13-17 Oct. 2003*, pages 168–174, 2003.

[115] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[116] Sandip Roy, Yan Wan, and Ali Saberi. A flexible algorithm for sensor network partitioning and self-partitioning problems. In *Algorithmic Aspects of Wireless Sensor Networks*, volume 4240 of *Lecture Notes in Computer Science*, pages 152–163. Springer Berlin / Heidelberg, 2006.

[117] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[118] Gerald Salton. *Automatic text processing*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[119] T. Saracevic and L. Covi. Challenges for Digital Library Evaluation. In *Proceedings of the ASIS Annual Meeting*, volume 37, pages 341–350, 2000.

[120] Somwrita Sarkar and Andy Dong. Community detection in graphs using singular value decomposition. *Physical Review E*, 83:046114, Apr 2011.

[121] P. B. Seddon, S. Staples, R. Patnayakuni, and M. Bowtell. Dimensions of information systems success. *Communications of the AIS*, 2(3es), November 1999.

[122] C. Shahabi, F. Banaei-Kashani, Chen Yi-Shin, and D. McLeod. Yoda: an accurate and scalable Web-based recommendation system. In *9th International Conference on Cooperative Information Systems*, pages 418–432. Springer-Verlag, 2001.

[123] Dan Shen, Jean-David Ruvini, and Badrul Sarwar. Large-scale item categorization for e-commerce. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 595–604, 2012.

[124] Alan F. Smeaton and Jamie Callan. Personalisation and recommender systems in digital libraries. *International Journal on Digital Libraries*, 5(4):299–308, 2005.

[125] Thornton Staples, Ross Wayland, and Sandra Payette. The Fedora project: An open-source digital object repository management system. *D-Lib Magazine*, 9(4), April 2003. `http://www.dlib.org/dlib/april03/staples/04staples.html` [last visited on December 15, 2013].

[126] Anselm Strauss and Juliet M. Corbin. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, September 1998.

[127] Anna Szczepanska. Research design and statistical analysis. *International Statistical Review*, 79(3):491–492, 2011.

[128] Tao Tao and ChengXiang Zhai. A mixture clustering model for pseudo feedback in information retrieval. In *Classification, Clustering, and Data Mining Applications*, Studies in Classification, Data Analysis, and Knowledge Organisation, pages 541–551. Springer Berlin Heidelberg, 2004.

[129] UCB DLI Team. UC Berkeley Digital Library Project. http://bscit.berkeley.edu/dlp.html [last visited on December 15, 2013].

[130] UCSB DLI Team. Alexandria Digital Library. http://www.alexandria.ucsb.edu/public-documents/metadata/metadata_ws.html [last visited on December 15, 2013].

[131] Chen Ting, Han Wei-Li, Wang Hai-Dong, Zhou Yi-Xun, Xu Bin, and Zang Bin-Yu. Content recommendation system based on private dynamic user profile. In *2007 International Conference on Machine Learning and Cybernetics*, volume 4, pages 2112–2118, 2007.

[132] S.L. Toral, M.R. Martinez-Torres, F. Barrero, and F. Cortes. An Empirical Study of the Driving Forces behind Online Communities. *Internet Research*, 19(4):378–392, 2009.

[133] University Corporation for Atmospheric Research. jOAI Software. `http://www.dlese.org/dds/services/joai_software.jsp` [last visited on December 15, 2013], 2002.

[134] Jian Wang, Badrul Sarwar, and Neel Sundaresan. Utilizing related products for post-purchase recommendation in e-commerce. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 329–332, 2011.

[135] Ryen W. White and Joemon M. Jose. A Study of Topic Similarity Measures. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 520–521, 2004.

[136] R. L. Williams and J. Cothrel. Four smart ways to run online communities. *Sloan Management Review*, 41(4):81–92, 2000.

[137] I. H. Witten and D. Bainbridge. *How to Build a Digital Library*. Morgan Kaufmann Publishers, San Francisco (CA), USA, 2003.

[138] H. I. Xie. Users' evaluation of digital libraries (DLs): Their uses, their criteria, and their assessment. *Information Processing and Management: an International Journal*, 44:1346–1373, 2008.

[139] Phillip M. Yelland and Eunice Lee. Forecasting product sales with dynamic linear mixture models. Technical report, Sun Microsystems, Inc., 2003. SMLI TR-2003-122, `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.196.4866&rep=rep1&type=pdf` [last visited on December 20, 2013].

[140] Zhiyong Zhang and Olfa Nasraoui. Mining search engine query logs for query recommendation. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 1039–1040, 2006.

Appendix A

IRB Approval Letter

We received approval from the Virginia Tech Institutional Review Board for conducting the focus groups on October 5, 2010.

VirginiaTech

Office of Research Compliance
Institutional Review Board
2000 Kraft Drive, Suite 2000 (0497)
Blacksburg, Virginia 24060
540/231-4606 Fax 540/231-0959
e-mail irb@vt.edu
Website: www.irb.vt.edu

MEMORANDUM

DATE: October 5, 2010

TO: Edward A. Fox, Monika Akbar, Yinlin Chen, Weiguo Patrick Fan

FROM: Virginia Tech Institutional Review Board (FWA00000572, expires June 13, 2011)

PROTOCOL TITLE: Ensemble Focus Group

IRB NUMBER: 10-431

Effective October 4, 2010, the Virginia Tech IRB Chair, Dr. David M. Moore, approved the new protocol for the above-mentioned research protocol.

This approval provides permission to begin the human subject activities outlined in the IRB-approved protocol and supporting documents.

Plans to deviate from the approved protocol and/or supporting documents must be submitted to the IRB as an amendment request and approved by the IRB prior to the implementation of any changes, regardless of how minor, except where necessary to eliminate apparent immediate hazards to the subjects. Report promptly to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.

All investigators (listed above) are required to comply with the researcher requirements outlined at http://www.irb.vt.edu/pages/responsibilities.htm (please review before the commencement of your research).

PROTOCOL INFORMATION:
Approved as: **Expedited, under 45 CFR 46.110 category(ies) 6, 7**
Protocol Approval Date: **10/4/2010**
Protocol Expiration Date: **10/3/2011**
Continuing Review Due Date*: **9/19/2011**
*Date a Continuing Review application is due to the IRB office if human subject activities covered under this protocol, including data analysis, are to continue beyond the Protocol Expiration Date.

FEDERALLY FUNDED RESEARCH REQUIREMENTS:
Per federally regulations, 45 CFR 46.103(f), the IRB is required to compare all federally funded grant proposals / work statements to the IRB protocol(s) which cover the human research activities included in the proposal / work statement before funds are released. Note that this requirement does not apply to Exempt and Interim IRB protocols, or grants for which VT is not the primary awardee.

The table on the following page indicates whether grant proposals are related to this IRB protocol, and which of the listed proposals, if any, have been compared to this IRB protocol, if required.

Date*	OSP Number	Sponsor	Grant Comparison Conducted?
10/4/2010	08272706	NSF	yes on 10/4/2010

*Date this proposal number was compared, assessed as not requiring comparison, or comparison information was revised.

If this IRB protocol is to cover any other grant proposals, please contact the IRB office (irbadmin@vt.edu) immediately.

cc: File

Appendix B

Invitation Email

The following email was sent to focus group participants.

Dear All,

Ensemble (www.computingportal.org) is a distributed portal for computing education. As part of this NSF-funded project, we are investigating how to fulfill information needs of our key target audience: the educators. As an educator in information system/information technology, we would like your participation in a focus group to help us better understand your needs.

Please indicate your availability to participate in a focus group session, using the following URL: http://www.doodle.com/ybkvinmbu4egkcb7

We will send you a separate email regarding the location of the focus group.

We want to thank you in advance for your consideration and willingness to help.

VT Ensemble team (Monika Akbar, Patrick Fan, Ed Fox)

Appendix C

Informed Consent Form

Informed Consent for Participants
in Research Projects Involving Human Subjects

Focus Group for Ensemble

Investigators: Edward A. Fox, Patrick Fan, Monika Akbar, Yinlin Chen

I. Purpose of this Research

The purpose of this research is to investigate how Ensemble can be effectively used by a certain group of educators. The goal is to identify areas where we can improve our services.

II. Procedures

We will conduct a focus group study that will be approximately one hour long. At the beginning of the focus group, each participant will be given an informed consent form. We will have a set of questions that will be asked to the participants. One of the investigators will act as a moderator who will initiate the questions and direct the course of the conversation. Other investigators will take written notes and will record the conversation.

III. Risks

There is very little, if any, risk associated with this study. The only risk could be presenting some opinion. Since the participants' data will be confidential, there will be no direct connection between a participant and any opinion s/he expresses.

IV. Benefits

The focus group will help us identify major areas that we need to improve in order to make the site more effective and useful to end user. The issues identified in this session will guide the Ensemble development team in the ongoing design phase of the website.

V. Extent of Anonymity and Confidentiality

Each participant will be assigned a random number when they sign the informed consent. This number will be used to keep the participants anonymous. The informed consents will be stored in a locked file cabinet in a secure research facility under the protection of the investigators.

We will record the conversation during the focus group. We will also take notes some times. The audio from focus groups and hand notes will be converted to digital text format. The notes will be destroyed by shredding the paper. The digital recordings and notes will be stored on an external drive which will be password protected and kept in a secure locked research facility. At no time will the researchers release identifying information to anyone other than individuals working on the project without your written consent. All data will be destroyed within 5 years of the experiment by either shredding the documents or erasing the external drive.

It is possible that the Institutional Review Board (IRB) may view this study's collected data for auditing purposes. The IRB is responsible for the oversight of the protection of human subjects involved in research.

VI. Compensation
No monetary compensation will be provided for participation in this study.

VII. Freedom to Withdraw
You are free to withdraw from a study at any time without penalty. If you choose to withdraw, you will not be penalized in any way. You are free not to answer any questions or respond to experimental situations that makes you uncomfortable. There may be circumstances under which the investigator may determine that a subject should not continue as a subject.

VIII. Subject's Responsibilities
I voluntarily agree to participate in this study. I have the following responsibilities: to let the experimenter know if I am feeling uncomfortable and need to take a break or leave the study.

IX. Subject's Permission
I have read the Consent Form and conditions of this project. I have had all of my questions answered. I hereby acknowledge the above and give my voluntary consent:

_____ Date:_____
Subject Signature

Should I have any pertinent questions about this research or its conduct, and research subjects' right, and whom to contact in the event of a research-related injury to the subject, I may contact:

Dr. Edward A. Fox (Faculty), Dr. Patrick Fan (Faculty), Monika Akbar, Yinlin Chen
Computer Science Department
Blacksburg, VA. 24061
{fox, wfan, amonika, ylchen}@ vt.edu

Dr. Ryder
Department Head, Computer Science Department
2202 Kraft Drive
Blacksburg, VA 24060
Ryder@cs.vt.edu
Phone: 540.231.8452

NOTE: Subjects must be given a copy (or duplicate original) of the signed Informed Consent.

David M. Moore
Chair, Virginia Tech Institutional Review
Board for the Protection of Human Subjects
Office of Research Compliance
2000 Kraft Drive, Suite 2000 (0497)
Blacksburg, VA. 24060
540.231.4991
MooreD@vt.edu

NOTE: Subjects must be given a copy (or duplicate original) of the signed Informed Consent.

Appendix D

Data Collection and Analysis

For the focus groups, we followed a two-step process of data collection and analysis. We identified key areas of Ensemble for further research, developed a protocol for conducting focus groups, conducted the focus groups. During the focus groups an audio recorder was used to capture audio. The audio files were transcribed and later used for further analyses.

Table D.1: Phases of data collection and analysis.

Data Collection	
System Review	Identified key areas of the Ensemble library for further research and development.
Protocol Development	Created a protocol and a set of questions for the focus groups.
Focus Groups	Conducted two focus groups at Virginia Tech. Each focus group was roughly one hour in duration.
Participants	Each of the 9 participants were Business faculty who teach computing to Business majors.
Data Analysis	
Transcription	The audio recordings were transcribed and combined with handwritten notes taken during the session to create a combined report of the two focus groups.
Coding	We identified repeated answers, patterns and behaviors in the transcribed data and in the report. These were coded based on the themes they represented.
Themes	The codes were used to identify emerging themes which were then used to develop and connect high-level codes about the prevalent practices on locating and using electronic resources, on creating active users in an educational DL.