Toward the Exploitation of Social Access Patterns for Recommendation

Jill Freyne
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland
jill.freyne@ucd.ie

Rosta Farzan
Intelligent Systems Program
University of Pittsburgh,
Pittsburgh, PA, USA
rosta@cs.pitt.edu

Maurice Coyle
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland
maurice.coyle@ucd.ie

ABSTRACT

The size and diversity of the Web has been the root cause of the poor performance of many retrieval systems, with little navigational support provided by many large online information repositories. The online information retrieval process across different repositories shares similarities with content access facilities and user behaviors even when containing inherently different content types. In this work, we introduce our social recommender system called ASSIST. The recommendation framework in ASSIST can be applied to any online information retrieval service with key information access components, search and browsing. ASSIST exploits multiple forms of social implicit feedback in order to generate well-informed user recommendations in the online information retrieval domain.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing method; H.5.3 [**Group and Organization Interfaces**]: Computer-supported cooperative work

General Terms

Experimentation, Human Factors

Keywords

Social access patterns, search, browsing, implicit feedback

1. INTRODUCTION

Due to the spiraling growth of the Internet [8], the process of locating relevant information is becoming more and more difficult for users. Retrieval systems find it close to impossible to index large portions of the web and even harder to decipher users' intentions from their poorly formed statements of information need (i.e. queries) [6].

Recent research has drawn the attention of researchers in the field to look into the exploitation of user interaction patterns as implicit feedback to drive retrieval. For example, work in the area of collaborative search shows how the exploitation of query submission and page selection information can improve the relevance of retrieved results for a community of searchers [10]; similarly positive results can be seen when historical browsing patterns are made visible to users as they browse through an information space [5]. Each of these techniques exploits a form of interaction history to make retrieval recommendations to users which cover a single aspect of the environment in which the user is situated, for example a search result-list or a Web page with multiple navigation options.

In this work, we seek to harness complementary interaction history data collected under social searching and social browsing paradigms and use them to recommend useful or interesting digital content to the users of any online information access service that provides both searching and browsing facilities. We will present the ASSIST, (**A**daptive **S**ocial **S**upport for **I**nformation **S**pace **T**raversal), social support engine, which encapsulates a generic framework for the deployment of integrated social search and social browsing support in an information access system, storing interaction history data and utilizing historical access patterns to make recommendations to users as they attempt to satisfy their information needs. We expand on previously reported work on ASSIST [4, 3] and introduce its new contextual recommendation functionality.

2. RELATED WORK

Over the last decade, social and collaborative filtering have been raised as important approaches to addressing the information overload problem. Social filtering techniques try to harvest the collective wisdom of a community to help individual users. The high cost of explicit feedback has pushed these techniques to very often use the implicit feedback such as the browsing activity, entered queries, or provided tags. I-SPY [10] is an example of a social search system which exploits community implicit interaction patterns to make recommendations in a search scenario. It promotes search results which have been selected in the past for a community of users and provides justification for its recommendations through the use of social icons. Lehikoinen et al [7] used a very similar approach in developing a meta search engine allowing implicit, anonymous, and asynchronous collaborative searches. Similarly they utilize the searches made by other network users in the past. The past search queries are then used to create a result relevancy indicator based on the

RecSys'07, October 19–20, 2007, Minneapolis, Minnesota, USA.

previous searches. Social browsing systems such as KnowledgeSea II [1],and CoFIND [2] exploit the number of visits to each resource as implicit feedback to discover community preferences. KnowledgeSea II combines users' annotations and time spent reading to enrich the feedback. CoFIND combines implicit feedback with users' explicit rating to collect stronger evidence of community preferences. QuickStep [9] is a research paper recommender which provides recommendation by recording the papers visited by each user. QuickStep uses supervised machine-learning techniques and an ontological representation to extract user preferences. To enrich community based recommendations, ASSIST aims to combine several sources of implicit feedback.

3. THE ASSIST ENGINE

The ASSIST engine has two core components, a search component and a browsing component (see Figure 1). Each component maintains a separate store of information access patterns for multiple communities of users of an information access system and both stores are exploited in an integrated fashion by the ASSIST engine to produce content recommendations. Briefly, the proxy-based architecture of the ASSIST engine intercepts user requests to an information repository coming from several possible browsing and searching entry-points or *hooks*, including search boxes, toolbars, result selections and user clicks.

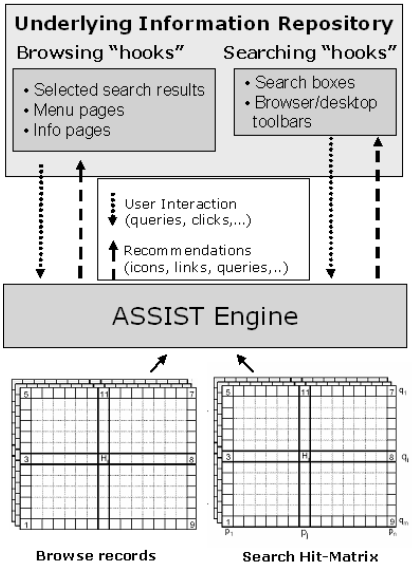

Figure 1: ASSIST engine system architecture

The ASSIST architecture can be adapted to any information repository which has a search and browsing facility. In previous work we illustrated our ideas using the ACM Digital Library. In this work we will refer to examples using a multimedia platform for serving up videos, YouTube (www.youtube.com) to illustrate how our recommendation strategies could be implemented in an alternative domain.

3.1 The Search Component

The search component maintains a store of all search activity encountered within the system. It records each submitted query and the subsequent result selections, thus capturing implicit relevance feedback from the community members. These search histories are stored in the so-called search *hit-matrix*, H. A hit-matrix is a grid-like data structure that stores query and result selection information for a given community and thus contains a set of user access patterns that reflect the collective preferences of the community. Each element of the hit-matrix, H_{ij}, stores a value v which indicates that page p_j has been selected for query q_i v times. v is updated each time p_j is reselected for query q_i.

3.2 The Browsing Component

The primary function of the browsing component is to collect and store browsing interactions with an underlying information repository. Two forms of browsing information are collected; simple browsing and contextual browsing. Simple browsing clicks are recorded when a user selects a link that is contained in a menu list. The newly-extended ASSIST considers two forms of contextual browsing, query-context browsing and page-context browsing. A page p_T is considered to have an associated page context p_C if p_C contains a hyperlink to p_T and a contextual click is recorded if a user follows the link.

A page p_T is seen to have a query context, q_C, if the path being navigated by the user started at a result-list for q_C and the user has navigated away from the result list. We make the assumption that a user's goal is still valid if they decide to navigate outwards from a result page but we do recognize that as the user navigates away from the result list, there is an increased chance that the user may have changed their original information goal. Thus the strength of the contextual relationship degrades with each click away from the result-list.

Two data structures make up the browsing component's information store. The simple clicks are stored in a database which for each community records the number of times the hyperlink is selected versus the number of times a hyperlink has been presented to a user. A second type of hit-matrix (BH, the *browse matrix*) records contextual click information. The browsing context c (query or content item) is paired with the resulting clicked page p_T and the cell representing the pair, BH_{cT} is incremented each time such a selection reoccurs. The browse matrix also stores the number of times the user had the option to select p_T but did not follow that path and the distance from the goal.

4. EXPLOITING INTERACTION PATTERNS

As described, the ASSIST engine stores several forms of user interaction activity information and exploits it to recommend content from an information repository in both searching and browsing situations. ASSIST bases its recommendations on the past interactions of community members, which can be thought of as implicit relevancy judgements by the users. Unlike other systems that monitor and recommend content within either a browsing or a searching context, ASSIST exploits both its search and browsing histories in both contexts. ASSIST has three strategies for delivering recommendations to users; they consist of augmenting links with social cues, context-based recommendations using mouseovers, and list reordering.

4.1 Social Cues

Graphical *social cues* are added throughout the interface

to links to highlight areas of interest and suggested paths through the space based on past interactions. Two social cue icons are used; a magnifying glass to depict search-related information and a trail of footprints for browsing information (see Figure 2). The *fill-level* of the icons represents the strength of the recommendation. The use of two icons allows users to determine whether the recommendation is based on search or browsing information or both. This is important since in certain scenarios the user may be more interested in one source than the other. For instance when engaged in a focussed search, information about an item's general browsing history may not be as important as the queries which have previously led to the item being viewed.

4.1.1 Search Icon

A content item, I, within a repository is augmented with the ASSIST search icon if previous users have accessed I using the repository's search facility. The icon's *fill-level* depicts the popularity of I amongst community members across search queries. That is, for each target query, q_T that led to I's selection, the *popularity*, $Pop_{search}(I, q_T)$ of item I for query q_T can be calculated by obtaining the number of times I has been selected in response to q_T as a percentage of the total number of selections across all items for q_T. Thus, the strength of a search-based recommendation for item I is the *average popularity* of I across all queries which have previously led to I being selected.

4.1.2 Browse Icon

The browse icon reflects the *browsing popularity* of a content item I within an information repository. The importance of the browsing component of a recommendation made by the ASSIST engine is calculated using 2 distinct measures of browsing popularity. The *simple browse popularity* of an item I, $Browse(I)$, is expressed by the number of times item I has been selected divided by the total number of times I has been displayed, in a browsing context. On the other hand the *contextual browse popularity* of I, $Browse_{cont}(I, I_C)$, provides a measure of how popular item I is *in the context of* the page or content item that the user is currently viewing, I_C, which we call the *context* of I. $Browse_{cont}(I, I_C)$ is calculated by dividing the number of times item I has been selected within the context of item I_C by the total number of times I has been displayed in the context of I_C multiplied by a weight which is a function of distance of the page from the goal page.

4.2 Contextual Recommendations

A user encounters a contextual recommendation as a result of mousing over a social icon. The mouseover displays either contextual query recommendations (in the case of the search icon) or contextual content recommendations(in the case of the browse icon). Each recommendation is clickable, allowing the user to retrieve the results of a recommended query or directly access some recommended content.

4.2.1 Query Recommendation

Mousing over a search icon reveals a list of recommended queries. Each recommendation corresponds to a query that has resulted in the selection of the target item in the past as detailed previously. The strength of each recommendation, and thus the ordering of the list, is based on each query's popularity $Pop_{search}(I, q_n)$ for item I.

4.2.2 Content Recommendation

Content recommendations or page recommendations are contained in the mouseovers of the browse icons. Each recommended item I has been selected by a community member directly after the item which the user is now viewing, I_C. The recommendation ordering is determined by the contextual browse popularity, $Browse_{cont}(I, I_C)$ detailed above. By following the recommendations users can skip a step in the trail followed by others. Figure 2 shows the recommendations made to a user based on the query "*happy*" and the search result "*happy feet*" which the user has moused over. The recommendations include the query "*happy feet*" and recommendations for other content "*happy feet penguin dance*" and "*happy feet remix*" which have been viewed directly after the search result "*happy feet*" clip.

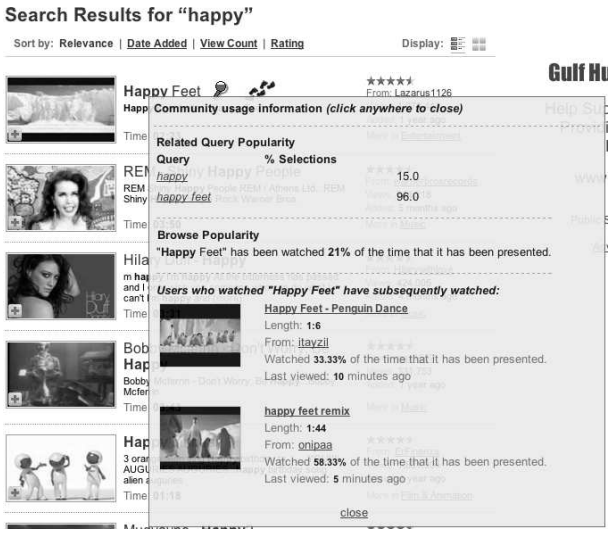

Figure 2: Result page from ASSIST YouTube.

4.3 List Reordering

ASSIST's most pro-active form of recommendation is obvious in the reordering of lists such as search result-lists and related item lists. These lists are reordered, promoting recommended items to users based on previous user interactions.

4.3.1 Search Result Lists

ASSIST exploits the Collaborative Web Search (CWS) technique [10] in order to make recommendations at search time. Briefly, at query-time, the ASSIST engine dispatches the user's target query, q_T to the search component, which generates a list of recommendations, R_A. In parallel, the standard list of results, R_S, returned by the underlying information access system is retrieved and reordered so that the top k recommendation candidates (typically, k=3) from R_A are promoted/inserted to the top of the search result-list.

In Section 4.1.1 we detailed how an items' popularity amongst community members across all search queries is calculated. ASSIST exploits a similar metric which concentrates on an items popularity for a specific target query in order to generate R_A for each search query. For a target query, q_T the *relevance*, $Rel_{search}(I, q_T)$ of item I is calculated by obtaining the number of times I has been selected

in response to q_T (and queries similar to q_T) as a percentage of the total number of selections across all items for q_T. Thus R_A contains a list of items I which have been selected in the past for q_T and is ordered by decreasing relevance. The browsing component's information store is accessed and browsing history information is retrieved and used to also provide the user with information about each result's browsing history before the final recommendations list is returned.

4.3.2 Related Pages List

Related item lists are reordered by recommending items which have been previously selected from the current context. Thus each related list is reordered based on the contextual browse popularity, $Browse_{cont}(I, I_C)$, of each I in the list. An example of this list reordering is seen in Figure 3 where related videos "*happy feet remix*" and "*happy feet penguin dance*" are promoted from their previous positions in the related video list.

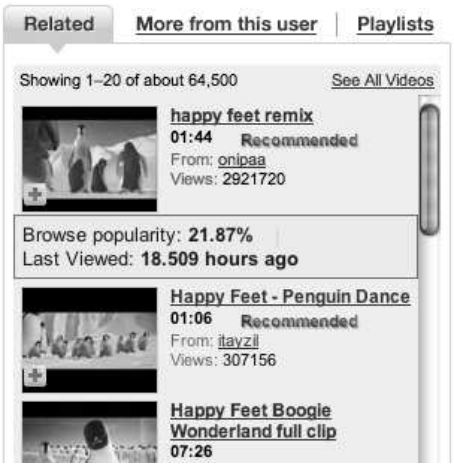

Figure 3: Reordered YouTube related videos list.

5. DISCUSSION AND FUTURE WORK

We have described a recommendation framework called the ASSIST engine that incorporates notions of social search and social browsing in an integrated, complementary fashion, to help users navigate through an information space by recommending content based on the actions of previous users of an information access system. The ASSIST engine is designed as a flexible solution which can be used to augment any online information repository that provides its users with both searching and browsing capabilities.

Live-user evaluations of the ASSIST engine deployed within the ACM Digital Library have shown the potential of the integrated technique for enhancing the overall experience of users as they hunt for information in a large repository, engaging in both browsing and searching activities (see [3, 4]). In this work, we are seeking to position the architecture as a more general framework, which can be applied to the task of leveraging historical interaction data to recommend digital content to users in any sphere, from academic papers and general interest encyclopedia-style content to multimedia objects such as videos.

In future work, we will perform live-user evaluations of this framework in a variety of digital repositories, beginning with a deployment over a very large repository of online video content. Evaluating the framework in very different environments should demonstrate that the synergy between online searching and browsing paradigms is not particular to any one information access medium and thus it can be effectively exploited in many different deployment settings. This will highlight the general applicability of the approach and will serve to confirm the hypothesis underlying the project, i.e. that by recommending content to users based on the searching and browsing activities of previous users, the overall user experience can be enhanced and users find interesting online content more often and with less effort.

6. REFERENCES

[1] P. Brusilovsky, G. Chavan, and R. Farzan. Social Adaptive Navigation Support for Open Corpus Electronic Textbooks. In *Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH-04*, pages 24–33, 2004.

[2] J. Dron, C. Boyne, and R. Mitchell. Footpaths in the stuff swamp. In *Proceedings of WebNet 2001 - World Conference on the WWW and Internet*, pages 323–328, Florida, USA, October 2001.

[3] R. Farzan, M. Coyle, J. Freyne, P. Brusilovsky, and B. Smyth. Adaptive Social Support for Information Space Traversal. In *Proceedings of the 18th ACM Conference on Hypertext and Hypermedia (In Press) 2007*, Manchester,UK, 2007. ACM.

[4] J. Freyne, R. Farzan, P. Brusilovsky, B. Smyth, and M. Coyle. Collecting Community Wisdom: Integrating Social Search and Social Navigation. In *Proceedings of the 2007 International Conference on Intelligent User Interfaces, January 28-31, 2007,*, pages 52–61, Honolulu, Hawaii,USA, 2007. ACM.

[5] X. Fu, J. Budzik, and K. J. Hammond. Mining Navigation History for Recommendation. In *IUI '00: Proceedings of the 5th International Conference on Intelligent User Interfaces*, pages 106–112, New York, NY, USA, 2000. ACM Press.

[6] S. Lawrence and C. L. Giles. Accessibility of Information on the Web. *Nature*, 400(6740):107–109, 1999.

[7] Lehikoinen, Juha, Salminen, Ilkka, Aaltonen, Antti, Huuskonen, Pertti, Kaario, and Juha. Meta-searches in peer-to-peer networks. *Personal and Ubiquitous Computing*, 10(6):357–367, October 2006.

[8] P. Lyman and H. R. Varian. How Much Information, 2003. Retrieved from http://www.sims.berkeley.edu/how-much-info-2003.

[9] S. E. Middleton, D. C. D. Roure, and N. R. Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. 2001.

[10] B. Smyth, E. Balfe, J. Freyne, P. Briggs, M. Coyle, and O. Boydell. Exploiting Query Repetition and Regularity in an Adaptive Community-Based Web Search Engine. *User Modeling and User-Adaptated Interaction*, 14(5):383–423, 2004.