A Community-Aware Search Engine

Rodrigo B. Almeida Virgílio A. F. Almeida

Department of Computer Science
Universidade Federal de Minas Gerais
Belo Horizonte, MG 31270-010 Brazil
{barra,virgilio}@dcc.ufmg.br

ABSTRACT

Current search technologies work in "one size fits all" fashion. Therefore, the answer to a query is independent of specific user information need. In this paper, we describe a novel ranking technique for personalized search services that combines content-based and community-based evidences. The community-based information is used in order to provide context for queries and is influenced by the current interaction of the user with the service. Our algorithm is evaluated using data derived from an actual service available on the Web, an online bookstore. We show that the quality of content-based ranking strategies can be improved by the use of community information as another evidential source of relevance. In our experiments, the improvements reach up to 48% in terms of average precision.

Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*human factors, human information processing*; H.3.3 [**Information storage and Retrieval**]: Information Search and Retrieval—*retrieval models*; H.3.5 [**Information storage and Retrieval**]: Online Information Services—*web-based services*

General Terms

Human Factors, Experimentation

Keywords

searching and ranking, data mining

1. INTRODUCTION

Typical search engines show identical results for a given query independent of the user or the situation in which the query is being issued [15]. This may not be suitable since: (i) users may have different information need, (ii) the information need of a single user may change through time and (iii) the relevance of each object retrieved is extremely dependent on the context in which the query is issued [16]. For instance, the relevance of the results for the query "jaguar" will certainly depend whether the user is currently seeking information about F-1 pilots or is interested on the animal from the jungles of Suriname.

Techniques for community identification have been extensively used to improve the quality perceived by users of search engines. Communities have been incorporated as a source of information for ranking algorithms and new applications such as automatic directory creation. Furthermore, community identification studies have proven to be of great value to researchers trying to increase their understanding of the information society [1, 11, 12].

This paper presents and evaluates a novel ranking technique that uses the combination of these evidential sources of relevance: the content of the objects being retrieved and the interest-based community of the user issuing the search. The theory of Bayesian belief networks [20] is used as the unifying framework of our approach since it naturally adapts to the problem of combining two sources of evidence in a single Information Retrieval (IR) model. A simple case study based on data collected from an actual service available on the Web is presented.

Interest-based communities are groupings of users or, more specifically, user interactions that share common interests. They are created using clickthrough data recorded in the form of user surfing behavior. Our approach to community identification in the Web [2] is different from previous work since the majority of them usually rely on the explicit information provided by the authors of the services in terms of the hyperlinked structure connecting the Web pages provided [11, 12].

Since we base our analysis on the user's perspective, we are allowed to infer communities even for sources that do not explicitly show relationships between the pieces of information provided. As the Web evolves and new kinds of services (e.g., streaming media, online gaming) are created there is an urgent need for algorithms designed to work based on evidences other than link information.

Our model is directly applicable to Web services providing search interfaces to the content they provide, but it can also be adapted and useful to searching the whole Web. Although information about accesses distributed over the Web is not available, any search engine can certainly infer interest from information about the queries and subsequent accesses to documents returned to them. This piece of information can be easily gathered through the use of off-the-shelf technologies [14].

Query contextualization is achieved by the juxtaposition of the current user interaction with a set of previous user interactions of all users in a way similar to collaborative filtering [6, 10, 14, 17]. Although we do not try to use user interactions other than the current one in order to characterize use interests when submitting a query, our technique can be easily extended to deal with sets of previous user interactions.

This paper is organized as follows: Section 2 introduces the method for identifying Interest-based communities. Section 4 shows how to combine the summaries with a content-based ranking technique using textual information as an indicative of relevance. Section 5 presents experimental results. Section 6 reviews related work. Section 7 discusses concluding remarks and future work.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-844-X/04/0005.

2. INTEREST-BASED COMMUNITY IDENTIFICATION

Usually, the information used by community identification algorithms is provided by the hyperlinked structure connecting Web pages [11, 12]. By modeling the Web as a graph and performing several operations on it, the authors were able so separate the Web in sets of related items.

Link information is provided on the creation of the page and is influenced by the author's view of the content provided and its relationship to pages. The content and outgoing links of a page represent a unique view about a subject, provided by its creator.

The incoming links to a page also represent unique views of that particular page. These views are not necessarily identical to the creator's, but still represent separated views of the same object. This distributed and uncoordinated nature of link creation is one of the main reasons for the success of community identification over the Web. In order to identify Interest-based communities for a service, similar entities for representing user accesses must provided.

Our approach is to model users and their interests in a graph and use the techniques already proposed for community identification as a means of identifying communities of interest. The structures we use to model user interaction as a graph are the *Session Interest-based Graphs* [2].

2.1 Session Interest-based Graphs

Session Interest-based Graphs, or simply SIGs, are centered on user sessions, i.e. a subset formed by the accesses issued by a user during a single interaction with a service. Since the scope of a session is restricted to a single user interaction, it is assumed that the objects contained in a session will be somewhat related and will mostly refer to a single interest.

Nodes in a SIG model the sessions of the service being analyzed into nodes of the graph. The weight of the edge connecting any two nodes representing sessions p an q is interpreted as their relationship and is denoted hereafter as $S[p, q]$. SIGs are an interesting modeling approach for they present characteristics that make them amenable to community identification such as: high clustering coefficient, small diameter and the existence of a giant strongly connected component.

In order to build representative SIGs not all requests made by the users are considered. Although we need to consider all user requests in order to correctly estimate user sessions, before an actual SIG is built, some of the requests are filtered.

The specific requests used in order to estimate session relationship are chosen according to business and institutional goals and will vary on an application basis. This distinction is made since some requests are more effective on characterizing user interest than others. For instance, in the case of an Online Radio the service provider can choose to use simply requests for songs coming from streaming media clients and ignores all other Web requests. In the case of a content provider all pages except the home page that is accessed by all of them might considered representative. From now on, we will refer to the remaining elements of the sessions simply as objects, independently of their type (i.e. page, streaming media video, etc).

Relationships between sessions are measured as cosine [5] between the vector representations of these sessions over the conceptual space formed by the objects requested by users. Despite the symmetry of the cosine measure, the SIGs are modeled as directed graphs so that algorithms for community discovery could be directly applicable to them.

3. COMMUNITY IDENTIFICATION

Several algorithms on the literature address the problem of community identification [1, 11, 12]. The HITS [12] is used, in this paper, mostly because it is a well-studied algorithm, derived from a well-known statistical technique, the *Singular Value Decomposition* (SVD). SVD has been successfully used in wide range of application varying from image segmentation to VLSI design [4]. Moreover, this technique has been proven to produce appropriate results when applied to a dataset with characteristics similar to the ones existent in the SIGs [4, 19].

The HITS algorithm was initially proposed as a method for improving the quality of searches over the Web. Its key idea is to identify hubs and authorities in a graph through a mutually reinforcing relationship existent between its nodes. This relationship may be expressed as follows: a good hub is a node that points to good authorities and a good authority is a node that is pointed to by good hubs. Thus, each node p, has associated with it an authority weight a_p, and a hub weight h_p. These weights form a ranking of the nodes ranging from good hubs/authorities, with high h_p/a_p values, to bad ones, with low h_p/a_p.

Let S denote the adjacency matrix representing the SIG from which one wants to identify communities (i.e. rows and columns represent sessions and entries represent similarity between the sessions) and a,h arrays storing authority and hub information for all the nodes. Therefore, authority and hub weights for the sessions can be iteratively computed as follows:

$$a = S^T h = S^T S a \quad (1)$$

$$h = S a = S S^T h \quad (2)$$

It has already been proved that, the authority and hub arrays, a and h, converge to the principal eigenvectors of $S^T S$ and SS^T respectively.

Subsequent work on HITS [12] showed that it can be used for discovery of communities if also non-principal eigenvectors of $S^T S$ and SS^T are considered as community descriptors. An implicit ranking of the communities can be derived by this method: the first non-principal eigenvectors identify the most important community over the nodes, the second non-principal eigenvectors explicit the second most important community over the nodes, etc.

The participation of node s in community c is given by the authority weight associated with each session in each community, denoted $a_{s,c}$. Let

$$S = U \Sigma V^T \quad (3)$$

be the SVD [8] for matrix S. Therefore, U defines the orthonormalized eigenvectors associated with the eigenvalues of SS^T (i.e. the hub weights), V defines the orthonormalized eigenvectors associated with the eigenvalues of $S^T S$ (i.e. the authority weights) and Σ is a diagonal matrix which elements are the square roots of the respective eigenvalues in non-decreasing order. Therefore, the authority values for each pair community/object can be retrieved from matrix V produced by the SVD of the SIG's adjacency matrix.

3.1 Community Summarization

For ranking purposes, a relationship between each community and the objects of its interest must be provided (i.e. a summary for each community). So far, we have only characterized relationships between user sessions and communities. These plus information about the objects requested in each session will serve as a starting point for the summarization of communities.

The sessions are split into three disjoint sets with respect to each community, the set of members, the set of non-members and the

rest of them. The set of members is constituted by the top-ranked sessions, with positive $a_{s,c}$ values. The non-members set is formed by the sessions occupying the lowest positions of the ranking, with negative $a_{s,c}$ values. The remaining sessions are included in a third set not considered through the rest of the summarization process.

The union between the set of objects requested in *member sessions* and the set of objects requested in the *non-member sessions* may not be disjoint. Therefore, for each community, we positively evaluate the objects requested in the *member sessions* and negatively evaluate the objects accessed in the *non-member sessions*. The weight associated with each pair object/session is calculated based on a *tf-idf* measure [5].

The interest of community c on object o is computed as follows:

$$w_{o,c} = \sum_{s \in m(c)} w_{o,s} - \sum_{s \in nm(c)} w_{o,s} \quad (4)$$

Where $m(c)$ represents the set of members of community c, $nm(c)$ the set of non-members of community c and $w_{o,s}$ the weight of object o in session s. The weights $w_{o,c}$ basically accounts for the requests made in the sessions existent in the members set minus the requests made in the sessions in the non-members set and can be used to evaluate the interest of each community on each object.

3.2 Identifying New Sessions

In order to be able to provide context for queries based on community information, we must also be able to effectively and efficiently assign a community for new sessions coming to the service. Suppose there exists a set of previous sessions already classified in communities and that those sessions are representative enough to capture most of user variability. The *Folding-in* method for SVD update can be used in order to compute authority weights for new sessions.

There exists some methods for SVD update in which the new included information contributes for the semantic latent structure identified, but there is a good trade-off between effectiveness and efficiency is obtained in using the *Folding-in* method and periodically reevaluating a new set of representative sessions. In this technique, the new information (i.e. new sessions) is seen as a projection into the already known semantic latent structure [8].

Let S' be an array representing the relationships between the sessions previously analyzed and the new one to be classified, denoted by s'. S' can be thought as a new column added to the original S matrix. The relationships represented by S' are also computed using the cosine measure (Section 2.1) [1].

Then, the authority weight associated with session s' in each community (i.e. $a_{s',c}$) can be computed as the projection of S' into the SVD space as follows:

$$a_{s',c} = S'^{T} U \Sigma^{-1} \quad (5)$$

where U and Σ are the matrices found in equation (3) for the set of sessions previously analyzed.

4. COMBINING CONTENT AND COMMUNITY INFORMATION

In this section we describe how to use community information to provide personalized searches for a content-based search engine relying on textual information. The novel Information Retrieval (IR) model proposed in this paper for combining textual and community information uses Bayesian belief networks as a unifying framework. Section 4.1 briefly introduces the Belief Network Model for Information Retrieval (IR) [5] and Section 4.3 shows how it can be used to combine community and content evidences in a single IR model.

[1]Note that in a real system, S' might have to be computed using partial information about a user session (e.g., in the case of a non-finished session).

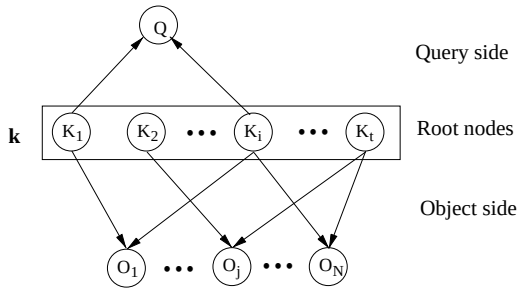

Figure 1: Example of a belief network for query Q specified using the terms K_1 and K_i

4.1 The Belief Network Model

A Bayesian Belief Network is a *Directed Acyclic Graph* (DAG) used to represent relationships among a set of random variables. Nodes, in this graph, are used to represent random variables and the directed edges connecting them portray their relationships. The amount of *belief* one has on the causal relationship between the random variables I and J, $i \rightarrow j$, is be modeled by the conditional probability $P(J = 1|I = 1)$ [20].

Bayesian Belief Networks are used in IR for they are a graphical formalism capable of representing independencies between variables of a joint probability distribution. The idea is that the known independencies among random variables of a domain are explicitly declared and that a joint probability is synthesized from this set of declared independencies [23]. Therefore, the combination can be done in a modular way so that it does not require that the pieces of information be based on the same principles nor does it require any modifications of them. Although other works have proposed to merge together several pieces of information (e.g. [6, 7, 10]) they are all binded to some sort of information modeling.

4.2 Modeling The Vector Space Model

Figure 1 shows a belief network that can be used as a framework to model all of the classical models for textual retrieval of objects, more precisely, the Boolean, the Probabilistic and the Vector Space models [5]. In these models the objects being retrieved and the queries are represented by the use of *index terms*. These are considered to be independent among themselves and are used as representations for elementary concepts of a conceptual space in which the documents are placed.

In the Belief Network Model, index terms are modeled by t nodes in the network, K_1 to K_t, where t is the number of unique index terms being considered. Vector **k** represents the conceptual space formed by them. Objects and queries are also mapped to nodes (e.g. O_j and Q).

The nodes enclosed in vector **k** are called root nodes. They are given this special treatment since they are the ones responsible for triggering the evaluation process. Given an instance of **k**, one can determine a similarity between the query and the objects.

Binary random variables are associated with each node in the network. It should always be clear whether we are referring to the the object/query/term, the respective node in the network or the random variable associated with it. Variable O_j is 1, denoted by o_j, to indicate that O_j is active and O_j is 0, denoted by $\bar{o_j}$, indicating that O_j is inactive. Analogously, variable Q is 1, denoted by q, to indicate that Q is active and Q is 0, denoted by $\bar{q}$, indicating that Q is inactive.

Although this is not the only interpretation existent, a similarity function between object O_j and query Q can be computed by calculating $P(O_j = 1 \mid Q = 1)$. This is done as follows:

$$\begin{aligned} P(O_j = 1 \mid Q = 1) &= P(o_j \mid q) \\ &= \eta \sum_{\mathbf{k}} P(o_j \mid \mathbf{k})\, P(q \mid \mathbf{k})\, P(\mathbf{k}) \quad (6) \end{aligned}$$

where η is a normalizing constant [24]. In spite of each node in the network being associated with a binary random variable, the varying degrees of relevance of the classical methods are achieved by the correct assignment of the conditional probabilities in equation (6). The final ranking is the summation of the similarities pointed out by each instance of the root nodes $\mathbf{k}$.

It has been show that equation (6) can be used to represent the classical models for IR. In our experiments (Section 5) we have used the Vector Space model [5] as an evidence of textual similarity between the objects and a query provided by the user. Here, we review how to use the belief network framework presented in the previous Section in order to compute the Vector Space ranking, or simply vectorial ranking, for a certain user query.

To compute the vectorial ranking, using the belief network introduced, proper probabilities $P(\mathbf{k})$, $P(q \mid \mathbf{k})$ and $P(o_j \mid \mathbf{k})$ must be provided. The probability $P(\mathbf{k})$, also called the prior probability of $\mathbf{k}$, associated with the root nodes is computed as follows:

$$P(\mathbf{k}) = \begin{cases} 1 & \text{, if } \forall_i\ t_i(Q) = g_i(\mathbf{k}) \\ 0 & \text{, otherwise} \end{cases} \quad (7)$$

where $t_i(Q)$ indicates whether the index term K_i was informed by the user in his query Q and $g_i(\mathbf{k})$ returns the value of the i-th random variable of $\mathbf{k}$ (i.e. K_i). Equation (7) establishes that only the states of $\mathbf{k}$ where the query terms are active will be considered.

Analogously, $P(q \mid \mathbf{k})$ is assigned to be:

$$P(q \mid \mathbf{k}) = \begin{cases} 1 & \text{, if } \forall_i\ t_i(Q) = g_i(\mathbf{k}) \\ 0 & \text{, otherwise} \end{cases} \quad (8)$$

Finally, $P(o_j \mid \mathbf{k})$ is computed by:

$$P(o_j \mid \mathbf{k}) = \frac{\sum w_{ij} \times w_{iq}}{\sqrt{\sum w_{ij}^2} \times \sqrt{\sum w_{iq}^2}} \quad (9)$$

where the summations are evaluated over the whole set of terms and w_{ij} and w_{iq} are tf-idf weights [5] relating the index term K_i with object O_j and query Q respectively.

By substituting equations (7), (8) and (9) in equation (6), we find that the similarity between object O_j and query Q, using the network of Figure 1, is evaluated as:

$$s(Q, O_j) \propto \frac{\sum w_{ij} \times w_{iq}}{\sqrt{\sum w_{ij}^2} \times \sqrt{\sum w_{iq}^2}} \quad (10)$$

that is proportional to the cosine of the angle between the vectors representing query Q and object O_j over the conceptual space formed by the index terms.

4.3 Combined Ranking

In order to include community information in the ranking, the network of Figure 1 needs to be extended. The extension we propose is considered to be modular since the inclusion of community information does not alter the way the original network was built. This is because community information is used as a way to provide context [15] for the textual query specified by the user.

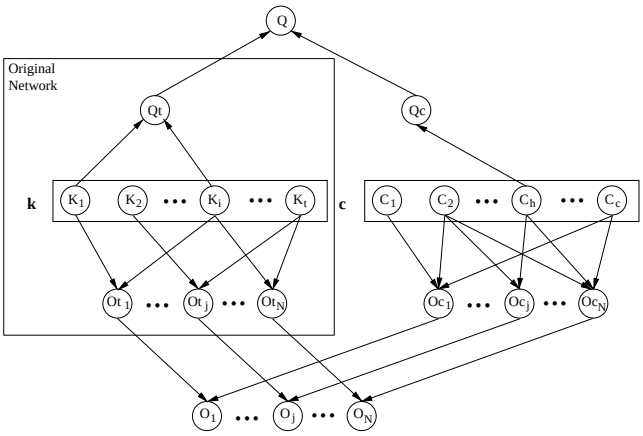

Figure 2: Modular extension of the original belief network for query Q using the fact that the user was characterized as pertaining to community C_h

Figure 2 shows the extended version of the original network. As can be noted, new nodes had to be added to the original network to cope with the inclusion of community evidence: (i) node Q_c is used to represent the community of the user that issued the query, (ii) a set of root nodes $\mathbf{c}$ that represents the set of communities being considered, (iii) a query node that is formed by the previous textual query and the context provided by the community, (iv) nodes to model the objects in terms of community relevance and (v) nodes to model the final results as a combination of the textual and community rankings. We further associate binary random variables with each of the new created nodes analogously to what have been done in Section 4.1.

The ranking of the objects provided by each community summary, used to describe the main interests associated with each community (Section 3.1), will be considered as the evidence of similarity between the objects and the users' interest during this search. The community weights of the objects were normalized within each community using decimal scaling in order to provide a value between 0 and 1 for this evidential source. Moreover, only the positive fold of the community summary is used (i.e. objects with negative $w_{o,c}$ values will not have their content-based ranking values affected).

Since the new communities we identified are based on a probably noisy data source (clickthrough data) and are also based on incomplete sessions (on-the-fly classification) we may expect a certain number of session misclassifications. The choice of considering solely the positive fold of community summaries is expected to reduce the loss in quality when a misclassification occurs since we only directly improve the content-based ranking of objects.

Analogously to Section 4.2, we can compute the final similarity of a contextualized query Q and object O_j by the value of $P(O_j =$

$1 \mid Q = 1)$. This is done as follows:

$$\begin{aligned} P(O_j = 1 \mid Q = 1) &= P(o_j \mid q) \\ &= \eta \sum_{\mathbf{k},\mathbf{c}} P(o_j \mid \mathbf{k},\mathbf{c})\, P(q \mid \mathbf{k},\mathbf{c})\, P(\mathbf{k},\mathbf{c}) \\ &= \eta \sum_{\mathbf{k},\mathbf{c}} P(o_j \mid \mathbf{k},\mathbf{c})\, P(q_t \mid \mathbf{k},\mathbf{c})\, P(q_c \mid \mathbf{k},\mathbf{c})\, P(\mathbf{k})\, P(\mathbf{c}) \\ &= \eta \sum_{\mathbf{k},\mathbf{c}} P(o_j \mid \mathbf{k},\mathbf{c})\, P(q_t \mid \mathbf{k})\, P(q_c \mid \mathbf{c})\, P(\mathbf{k})\, P(\mathbf{c}) \quad (11) \end{aligned}$$

As noted before, this extension is modular, therefore, $P(\mathbf{k})$, $P(q_t \mid \mathbf{k})$ are analogous to the probabilities assigned in Section 4.2. We need, therefore, to define $P(\mathbf{c})$, $P(q_c \mid \mathbf{c})$, $P(o_j \mid \mathbf{k},\mathbf{c})$. The prior probabilities $P(\mathbf{c})$ are computed as:

$$P(\mathbf{c}) = \begin{cases} 1 & \text{, if } \forall_i\ c_i(Q_c) = g_i(\mathbf{c}) \\ 0 & \text{, otherwise} \end{cases} \quad (12)$$

where $c_i(Q)$ indicates whether community C_i is the one query Q_c belongs to and $g_i(\mathbf{c})$ returns the value of the C_i random variable. Moreover, we establish that only one community can be specified as context for a given query. Although we define the properties of the network in a way that the context of a query is composed of only one community, this framework is general enough to allow extensions that consider more than one community as a context for a content-based search technique.

We set $P(q_c \mid \mathbf{c})$ analogously as:

$$P(q_c \mid \mathbf{c}) = \begin{cases} 1 & \text{, if } \forall_i\ c_i(Q_c) = g_i(\mathbf{c}) \\ 0 & \text{, otherwise} \end{cases} \quad (13)$$

At last, we assign $P(o_j \mid \mathbf{k},\mathbf{c})$ considering that it will use disjunctive *or* operation to provide the final ranking:

$$\begin{aligned} P(o_j \mid \mathbf{k},\mathbf{c}) &= 1 - [(1 - P(Ot_j \mid \mathbf{k},\mathbf{c})) \times (1 - P(Oc_j \mid \mathbf{k},\mathbf{c}))] \\ &= 1 - [(1 - P(Ot_j \mid \mathbf{k})) \times (1 - P(Oc_j \mid \mathbf{c}))] \quad (14) \end{aligned}$$

where $P(Ot_j \mid \mathbf{k})$ is analogous to $P(o_j \mid \mathbf{k})$ in Section 4.2 and $P(Oc_j \mid \mathbf{c})$ can be computed by:

$$P(Oc_j \mid \mathbf{c}) = \begin{cases} sw_{j,c} & \text{, if } w_{j,c} > 0 \\ 0 & \text{, otherwise} \end{cases} \quad (15)$$

where $sw_{j,c}$ is the scaled weight associated with object O_j in community c.

By substituting equations (7), (8), (9), (12), (13), (13), (14), (15) in equation (11) we find that the similarity of object O_j and query Q_t constrained to context Q_c can be computed as follows:

$$s(Q, O_j) \propto \begin{cases} 1 - [(1 - c(Q_t, j)) * (1 - sw_{j,c})] & \text{, if } w_{j,c} > 0 \\ c(Q_t, j)) & \text{, otherwise} \end{cases}$$

where $c(Q_t, j)$ is the cosine of the angle between the vectors describing textual query Q_t and object O_j (equation (10)).

5. EXPERIMENTAL RESULTS

In this section we will present the dataset used in this work and also the results for the combination of textual and community information in the final ranking. Section 5.1 reviews the data collected from a service available on the Web that was used to test our approach. Section 5.2 shows the analysis of the results produced by the novel ranking technique proposed.

5.1 Dataset

The dataset we used, in this paper, was collected from a real online bookstore service. The high-level requests used to define user interests are the ones for information about books. In each of these requests we were able to correctly identify the referred books. Dealing with a search engine for books of an online bookstore is interesting for there are no explicit links or relationships among them. Therefore, we are able to notice the real advantage of new algorithms designed to work with evidences other than link information for the Web.

The full dataset comprises two weeks of accesses to the service, collected from August 1st to August 14th of 1999. This service is an e-commerce company, based in the US, specialized on Computer Science literature, operating only on the Internet. In the period recorded, the bookstore received 3.5 million requests, 87,000 of which were requests for information about books, such as: authors, price, category and reviews.

This dataset was split into two others: a training and a test dataset. The former comprises the first week of data collection and recorded 1.7 million requests with 26,000 sessions with at least one request for information about books identified. This dataset was used in order to estimate communities of users accessing this service.

One of the benefits of our approach allows us to decide the best number of communities based on the eigenvalues associated with the eigenvectors describing each community [2]. For this paper we arbitrarily identified the top 10 communities (i.e. the ones with higher eigenvalues) so that the number of communities is small enough to be evaluated by humans. The communities identified were named from C_1 to C_{10}. The number of significant communities for this dataset lies somewhere between 25-30 communities.

The latter dataset comprised the second week of data collection, received 1.5 million requests and had 17,000 sessions with at least one request for information about books. One of the main drawbacks associated with methods that use information from other users to characterize interest of other users based on similarity patterns is the problem of objects that have never been accessed before or have been accessed rarely. A first analysis of the number of the sessions that made requests for books might lead to erroneous conclusion that the session in the test set received less requests for books than the training one. What really happens is that we could only consider in the test set the books that had already been requested in the training set. Since we use community information as contexts for queries, the only drawback of our approach is that these objects will never have their content-based ranking altered. This follows a similar approach to the one taken in collaborative filtering area [6, 17].

Each of these 17,000 sessions identified in the test dataset was classified into the top 10 communities identified from the training dataset. The method we used is described in Section 3.2.

We found, on the period recorded, a total of 3,027 books for which requests were made. Information such as the categories that each book belongs to and a summary of each were collected from the Amazon [2] online store. The textual information considered in the content-based ranking technique is this summary describing the books.

5.2 Results

To be of practical use, we must show that our technique can be used to effectively model new sessions based on a latent structure identified from sessions previously analyzed. The first experiment we conducted was to: (i) find the best and worst communities for

[2] http://www.amazon.com

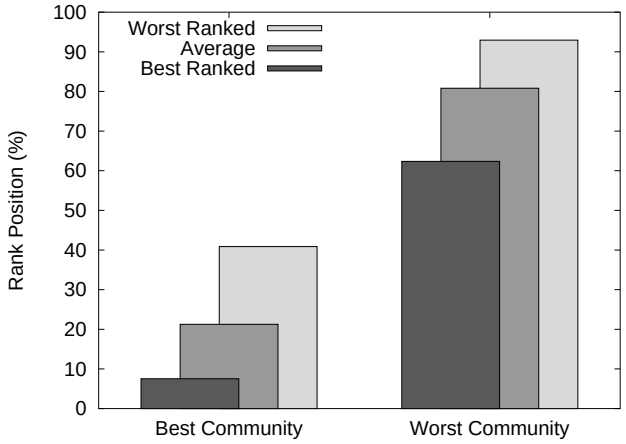

Figure 3: Normalized rank positions for the best ranked object, the worst and the average for all objects averaged over all queries considering the community rankings

each session of the test dataset (i.e. respectively the ones with highest and lowest $a_{s,c}$) and (ii) to analyze the positions in which the objects accessed in it appeared on these two community summaries.

Figure 3 shows the normalized ranking position for the best, the worst and the average object position on the best and worst community summaries averaged over the whole set of new sessions. In the normalized ranking, the value 0 indicates that the object was found to be on the first position of the ranking while the 100 value indicates that the object was located at the last position of the ranking.

Suppose we were analyzing a dataset with ten objects and session s accessed objects o_1, o_2 and o_3. Moreover, suppose that the best and worst communities for s are c_b and c_w respectively. If the summary of c_b object o_1 appeared in the third position, o_2 appeared in the sixth position and o_3 appeared in eighth position, the normalized ranking position of the best ranked object (Best Community's Best Ranked bar) would be 30%. The normalized ranking position for the average object (Best Community's Average bar) would be 50% and the normalized ranking position for the worst ranked object would be 80%.

As can be noted the the majority of objects that co-occurred in these sessions really appeared on the first half of the ranking provided by the best community and on the second half of the ranking provided by the worst one. Therefore, the values $a_{s,c}$ can be used to characterize the participation of an object in a certain community and communities are truly able to characterize users with different interests since the two groups of bars diverge.

Information about the categories that each book belongs to was used to show that communities really separat the users in groups with different interests. The weight of each book, computed as proposed in Section 3.1, was accumulated for each pair category community. Table 1 shows the qualitative analysis of subjects covered by communities C_5 and C_{10}. An in-depth look of all top 10 communities identified can be seen in [2]. As can be noted, these two communities represent sets of users with distinct interests. While community C_5 is specifically interested in database technologies, community C_{10} aggregates users interested in low level questions basically related to Unix-like operating system's issues and the algorithm was able to identify these two distinct groups without the use of any a-priori information.

	Best-ranked categories	**Worst-ranked categories**
C_5	Specific Databases; Database Management Systems; Database Design	Certification; Networking; Software Design
C_{10}	Networking; Linux; Unix Operating Systems	Certification; Specific Databases; Microsoft Development

Table 1: Qualitative analysis for communities C_5 and C_10

Table 2 shows the top 10 books retrieved for a broad-topic query, *"system administration"*, using the vectorial ranking alone and its combination with communities C_5 and C_{10}. The vectorial ranking returns generic results with respect to the query covering several aspects of it, namely, operating systems, networking, database and ERP systems. The results found by the combination of textual and community information show that community information is useful in focusing on a specific subject. From the results returned for community C_5 seven of results are relevant for this community while only one could be found using the classical vectorial ranking. For the combination with community C_{10} we found a similar relationship of 8 to 5. Several other broad-topic queries showed similar results.

This experiment shows the utility of our methodology to cope with the problem of broad-topic queries in the Web [15]. These queries are usually composed of a small number of index terms that generally refer to several knowledge areas. By providing context, in terms of the communities identified from user accesses, we are capable of producing results that exceed the quality of the ranking of typical content-based techniques.

We also performed another experiment to show that our algorithm is capable of providing better rankings given that community information is available. A set of queries and relevant sets were synthetically created. These queries were based on data from the logs recorded for the online bookstore. The process of query creation was done as follows:

for all $s \in T$ **do**
  if $(|O(s)| > 1)$ **then**
    for all $o \in O(s)$ **do**
      create a new query: q;
      make $q =$ title of book o;
      create a set of relevant objects for this query: r;
      make $r = O(s) \setminus \{o\}$;
      associate the best and worst communities for s with q;
    end for
  end if
end for

where T is the set of sessions identified in the test dataset and $O(s)$ the set of objects accessed in session s. Using this procedure we were able to create 23,300 queries.

These queries were submitted to a database formed by the objects accessed in the test dataset. For each query we considered the usual content-based ranking and two variations of our approach. For the community-aware search engine experiment, we considered the communities in which the session that originated each query were ranked best and worst. For instance, if query q was originated from session s, we would process three queries: (i) the pure textual query based solely on content, (ii) the content-based query in conjunction with community c_b and (iii) the content-based query in conjunction with community c_w.

(a) Vectorial Ranking

1	Microsoft Windows NT 4.0 Administrator's Pocket Consultant
2	Essential Windows NT System Administration
3	UNIX System Administration Handbook (2nd Edition)
4	Oracle8 Administration and Management
5	AIX Version 4: System and Administration Guide
6	Zero Administration for Windows
7	HP-UX 11.x System Administration "How To" Book (2nd Edition)
8	SAP R/3 System Administration : The Official SAP Guide
9	Essential System Administration
10	The Cna/Cne Study Guide: Intranetware Edition (Certification Series)

(b) Vectorial & Community C5

1	Oracle Performance Tuning Tips and Techniques
2	Oracle Certified Professional Application Developer Exam Guide
3	Oracle8 Administration and Management
4	Oracle Database Administration: The Essential Reference
5	Sybase Dba Companion
6	AIX Version 4: System and Administration Guide
7	Oracle8 Advanced Tuning & Administration
8	Oracle8: A Beginner's Guide
9	HP-UX 11.x System Administration "How To" Book (2nd Edition)
10	The Linux Kernel Book

(c) Vectorial & Community C10

1	UNIX System Administration Handbook (2nd Edition)
2	The Linux Kernel Book
3	Operating System Concepts, 5th Edition
4	HP-UX 11.x System Administration "How To" Book (2nd Edition)
5	Linux: Companion for System Administrators
6	Microsoft Exchange Server in a Nutshell: A Desktop Quick Reference
7	Windows NT User Administration
8	UNIX System V: A Practical Guide (3rd Edition)
9	Microsoft Windows 2000 Beta Training Kit
10	Design of the Unix Operating System

Table 2: Results for the query *"system administration"*

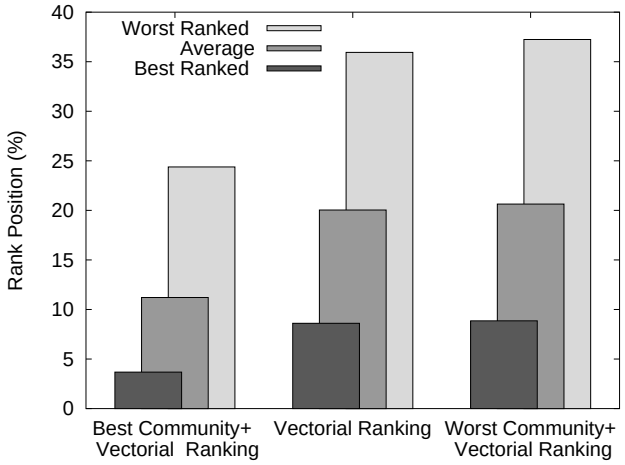

Figure 4: Normalized rank positions for the best ranked object, the worst and the average for all objects averaged over all queries considering the combination of the vectorial ranking with the community rankings

Figure 4 shows the normalized ranking position for the relevant objects averaged over all queries using the content alone and the combination with both the best and worst communities. On average, we get an improvement of 80% in terms of normalized ranking position when we combine textual information with the best community summary.

For the best ranked object the improvement is the highest, that means that the users are able to see the first relevant result for a given query in advance when we aggregate information about communities on the ranking technique. The same happens for the worst ranked object and this suggests that we might find better precision values for 100% of recall.

Figure 5 shows precision $\times x$ recall plots for the vectorial ranking and its combination with the best community. Although this synthetic query dataset is not appropriate for precision $\times x$ recall analysis we are able to see that the combined approach presents better precision values than the content-based technique alone. The improvement reaches up to 48% in terms of average precision. On average the number of relevant results for each query is too small since session sizes are small. Therefore, the relevant sets are underestimated and little perturbations on the ranking position of a single object causes great change on the associated precision value. Moreover, since queries are based on titles of books that generally contains many generic terms, the precision values we identify are quite low.

It is also interesting to note that the combination of textual information with the worst community (Figure 4) does not alter significantly the results. This is achieved since objects with negative $w_{j,c}$ values do not change the content-based ranking function. This characteristic is important since, in most of the times, the information for computing the actual community of the user will be based on partial information and, therefore, will not be ideal.

6. RELATED WORK

There exists some related work in web log mining [3, 9, 18, 21]. Our work differs from those in the following ways: i) Our method makes use of high-level application-oriented information (e.g., requests for books or the songs), to detect common interests among users, instead of looking for patterns of user's accesses; ii) Some

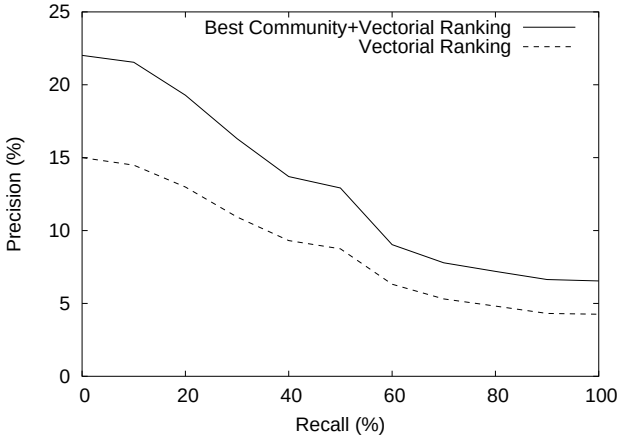

Figure 5: PrecisionxRecall values for the combined ranking using the best community

methods use simple object relationships [21], while other methods consider the order at which requests are made [3, 9] (e.g., using Markov models as a framework for modeling user interactions). Those methods would not work for our high-level datasets since most of the sessions have small sizes, 1 or 2 requests at most. iii) Our approach differs from classical clustering techniques, for we try to identify densely connected user sessions that may not account for much of the whole dataset while clustering techniques try to split the full dataset into a few subsets.

There has also been some work on the combination of rankings using a Bayesian approach. In [24] the authors combined textual information of a Web page with the one induced by its hyperlink connections. Our work differs from the previous in two ways: (i) we use implicit information provided by user behavior in order to create a graph structure capable of representing relationships between user interactions. Although each user behaves independently from others we were able to find structures that are similar to other ones explicitly created (e.g. small-worlds), (ii) the modeling effort on their work and ours is different since the hyperlink information used by then is somewhat related to the query terms while in our approach we model an evidential source that is independent from the textual information provided by the user in terms of his query.

A Bayesian model similar to the one used in [24] was also proposed in [25]. Although the modeling in these two works are slightly different it has already been shown that they exhibit similar expressive power.

In [10], the authors also propose a way to combine two information sources , the link-based information and the contents of the pages, in a single model. The approach taken by the authors is different from ours and from the others that are based on Bayesian Networks on the sense that this work tries to unify both sources using a single principle to evaluate them. They use variants of Latent Semantic Analyses (LSA) in order to evaluate the texts and links of the documents and then combine the results easily since the analyses were made based on the same principled manner. Although the combination produces good results it is rigidly connected to LSA.

The previous personalized search techniques for the web, such as [13, 16, 22], have mostly focused on how to gather information from a single user in order to provide context for his searches. Generally they are based on the creation of user profiles and make use of explicit actions from the users. In our approach we are able to combine information about several users together since the analysis is made in the servers and we are able to transparently provide context for the queries. Another difference from our approach to the others is that we only considered, so far, the actual interaction of the user with the service what means that we do not have to analyze changes on user interest but are subjected to less information for contextualization purposes.

In the area of recommendation algorithms several authors have shown the benefits of combining several sources of information in a single result as a way to improve the quality of the systems [6, 7, 14, 17]. In [6] the authors use *Ripper*, a rule induction system, as a means to learn a retrieval function that binary classifies movies with respect to a user (e.g. like or dislike). In [7] the authors use several textual information sources in order to recommend technical papers to be conference reviewers. Although their analysis is based on the same several sources of the same type they have shown that the combination of several information sources produced results that were better than the ones produced by the single sources alone. In [17] the authors have shown that one can use information from the objects (e.g., text) in order to enhance the quality of the recommendations made by traditional collaborative filtering approaches based on previous user relevance judgments.

The technique introduced in [14] uses clickthrough data in order to improve the quality of searches on the Web. The main difference between his work and ours is that he is only able to enhance the quality of the searches to a single user or groups of user that share common interests. Besides, the author works with information coming from the answers of queries to search engines and builds upon relative preferences between the documents returned. Our approach uses information from the whole sets of users to first characterize the user interest and then uses this information as context for subsequent queries of this user.

7. CONCLUDING REMARKS

Personalized search services will play a major role on the Web in the near future. In this type of application, a search for a certain textual query may return different results depending on context information (e.g. the user issuing the query, his current interests). In this paper we propose and evaluate a novel ranking technique that uses community information as a new evidential source for providing personalized ranking. Our approach is to use communities as contextualization cues for the queries.

The framework proposed is general enough to allow the use of any of the classical models for content-based information retrieval and of most of community identification algorithms, as long as summaries for the communities can be produced. In the experiments conducted in this paper we were able to provide an improvement of 48% in term of average precision, and even better results if we were to consider the position occupied in the ranking by the relevant objects for each query.

As future work on this area we envision: (i) adaptation of the framework for other alternative techniques for content-based ranking and community identification algorithms; (ii) experimentation with a real real (operational and interactive) system to demonstrate the effect of our method; (iii) the experimentation with partial session information as a way to characterize user interests; (iv) the expansion of the framework to work with more than one community as an evidential source for relevance. In order to tackle problems that might occur while using partial session information. The use of multiple communities may benefit from the fact that we have a value $a_{s,c}$ describing the participation of s in each community c.

Although the synthetic query dataset used in this paper is useful in demonstrating the ability of the combination proposed to im-

prove the quality of the searches, more detailed studies could be conducted if a better query dataset were available.

8. ACKNOWLEDGMENTS

The authors would like to thank the anonymous service owners and operators for enabling this research to proceed by providing us access to their logs. This work has been partially supported by a number of grants from CNPq-Brazil.

9. REFERENCES

[1] L. A. Adamic. *Network Dynamics: The World Wide Web*. PhD thesis, Stanford University, 2001.

[2] R. B. Almeida and V. A. F. Almeida. Design and evaluation of a user-based community discovery technique. In *Proceedings of the 4th International Conference on Internet Computing*, pages 17–23, 2003.

[3] C. R. Anderson, P. Domingos, and D. S. Wield. Relational Markov models and their application to adaptive web navigation. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 143–152, 2002.

[4] Y. Azar, A. Fiat, A. R. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *The 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 619–626, 2001.

[5] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[6] C. Basu, H. Hirsh, and W. W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720, 1998.

[7] C. Basu, H. Hirsh, W. W. Cohen, and C. G. Nevill-Manning. Technical paper recommendation: A study in combining multiple information sources. *Technical Paper Recommendation: A Study in Combining Multiple Information Sources*, 14:231–252, 2001.

[8] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.

[9] I. Cadez, S. Gaffney, and P. Smyth. A general probabilistic framework for clustering individuals and objects. In *Proceedings of the the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 140–149, 2000.

[10] D. A. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. *Advances in Neural Information Processing Systems*, 13:430–436, 2000.

[11] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[12] D. Gibson, J. M. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the 9th Conference on Hypertext and Hypermedia*, pages 225–234, 1998.

[13] E. J. Glover, S. Lawrence, M. D. Gordon, W. P. Birmingham, and C. L. Giles. Web search – your way. *Communications of the ACM*, 44(12):97–102, 2001.

[14] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, 2002.

[15] S. Lawrence. Context in web search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000.

[16] F. Liu, C. Yu, and W. Meng. Personalized web search by mapping user queries to categories. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 558–565, 2002.

[17] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 187–192, 2002.

[18] G. Paliouras, C. Papatheodorou, V. Karkaletsis, C. D. Spyropoulos, and V. Malaveta. Learning user communities for improving the services of information providers. In *European Conference on Research and Advanced Technology for Digital Libraries*, 1998.

[19] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: a probabilistic analysis. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems*, pages 159–168, 1998.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1997.

[21] M. Perkowitz and O. Etzioni. Adaptive web sites: Automatically synthesizing web pages. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 727–732, 1998.

[22] J. E. Pitkow, H. Schütze, T. A. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. M. Breuel. Personalized search. *Communications of the ACM*, 45(9):50–55, 2002.

[23] B. Ribeiro-Neto and R. Muntz. A belief network model for IR. In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, 1996.

[24] I. R. Silva, B. Ribeiro-Neto, P. Calado, E. Moura, and N. Ziviani. Link-based and content-based evidential information in a belief network model. In *Proceedings of the 23rd ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, 2000.

[25] H. R. Turtle and W. B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9:187–222, 1991.