Available online at www.sciencedirect.com
ScienceDirect
Journal of Network and Computer Applications 30 (2007) 1059–1083
Journal of NETWORK and COMPUTER APPLICATIONS
www.elsevier.com/locate/jnca

# Usage derived recommendations for a video digital library

Johan Bollen[a,b,*], Michael L. Nelson[b], Gary Geisler[c], Raquel Araujo[b]

[a]*Research Library, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[b]*Department of Computer Science, Old Dominion University, 4700 Elkhorn Avenue, Norfolk, VA 23529, USA*
[c]*School of Information, University of Texas, Austin, TX 78712, USA*

Received 14 December 2005; accepted 14 December 2005

**Abstract**

We describe a minimalist methodology to develop usage-based recommender systems for multimedia digital libraries. A prototype recommender system based on this strategy was implemented for the Open Video Project, a digital library of videos that are freely available for download. Sequential patterns of video retrievals are extracted from the project's web download logs and analyzed to generate a network of video relationships. A spreading activation algorithm locates video recommendations by searching for associative paths connecting query-related videos. We evaluate the performance of the resulting system relative to an item-based collaborative filtering technique operating on user profiles extracted from the same log data.


*Keywords:* Usage analysis; Recommender systems; Video; Open Video Project

*Corresponding author. Research Library, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. Tel.: +1 505 606 0030.
*E-mail addresses:* jbollen@lanl.gov (J. Bollen), mln@cs.odu.edu (M.L. Nelson), raraujo@cs.odu.edu (R. Araujo).
*URL:* http://public.lanl.gov/jbollen.

1084-8045/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2005.12.009

## 1. Introduction

Recommender systems can provide highly valuable services for large-scale Digital Libraries (DLs) (Smeaton and Callan, 2001); they aid users in finding relevant information and simulate an important social dimension of information seeking behavior (Twidale et al., 1997) namely that of users advising other users. This functionality becomes even more important where multimedia collections are concerned. We have all experienced the frustration of searching for a song whose "metadata" or "content" such as the title, author name or lyrics has eluded us, and then resorting to whistling fragments (Hu and Dannenberg, 2002) to confused record store clerks.

In spite of the increasing prevalence of DLs which store large-scale, mixed multimedia collections, retrieval and recommender systems for such DLs have remained quite rare even though this is where they seem to matter most. This situation may result from the lack of text content or rich metadata often associated with multimedia documents. Traditional information retrieval (IR) methods are largely dependent upon term indexing (Grossmman and Frieder, 1998) and thus have little applicability to multimedia collections unless metadata is generated by human indexers or image analysis techniques (Sun et al., 2003; Shyu et al., 2003), procedures which can be prohibitively expensive for large collections (Ioannou et al., 2000).

The Open Video Project (`www.open-video.org`) is a prime example of this issue. The Open Video DL provides access to a large collection of MPEG-1, MPEG-2, MPEG-4, and QuickTime formatted videos that are freely available for download. Although the videos have descriptive metadata, it is mostly manually generated and seldom describes a video's rich content and structure. Users can search on video title, abstract, description, type, format and numerous other metadata fields, but this represents a highly restricted interface to discover resources in a highly heterogeneous collection of videos. A recommender service that aids users to discover videos similar in content to the ones they or others appreciate would be highly valued both by Open Video users and its creators.

Due to the limited availability of text content and metadata, recommender services for commercial multimedia collections, e.g. `amazon.com`, are commonly implemented by employing Collaborative Filtering (CF) systems. CF was introduced in the early 1990s as a simulation of word-of-mouth recommendations (Shardanand and Maes, 1995) and has found widespread applications in commercial systems (Konstan et al., 1997; Schafer et al., 1999). Rather than operating on document text content or metadata, CF systems issue recommendations on the basis of previously stored user profiles which may consist of past purchases or submitted product ratings. First, user similarities are determined on the basis of whether users have similar profiles, i.e. whether they purchased similar items or submitted similar ratings. Second, recommendations are generated for an individual user by selecting items that are present in the profiles of similar users, but absent from their own. This procedure amounts to being recommended a

particular album by friends whose record collections indicate they have similar musical tastes.

CF variations have been proposed on the basis of hybrid systems which employ text matching techniques (Torres et al., 2004) and Latent Semantic Indexing to uncover prototypical user interest profiles and user communities (Hofmann, 2004). Sarwar et al. (2000, 2001) propose to determine item, rather than user similarities, from user profiles and apply this data to item recommendations. Nevertheless, a feature characterizing most CF systems is that they operate on representations of user behavior patterns, and therefore do not require text content or metadata and are therefore applicable to heterogeneous multimedia collections.

CF systems have proven widely applicable in commercial settings, but there exist a number of issues which hamper their deployment in large, multimedia DLs:

*Privacy expectations*. Unlike commercial or retail services, many DLs operate under stringent privacy requirements. Most DLs are deliberately not instrumented to record user profiles containing user names, addresses, preferences, credit card numbers or a history of past access patterns. For example, many DLs maintained by the U.S. Government are prohibited from collecting data about individual download behavior, cf. The Privacy Act of 1974, 5 U.S.C. § 552a. In addition, some DLs have recently adopted policies to destroy individual user records to prevent later analysis for law enforcement purposes (McFall and Schneider, 2003).

*Lack of persistent profiles*. DL user interests, particularly with regards to multimedia content, are highly ephemeral. They are not so much focused on stable interests than short-lived information needs. A DL user may continuously shift roles and interests over the course of a day, e.g. download personal literature before work and professional books for the remainder of the day. Systematically maintained user profiles may be confounded by such behavior.

*User resistance*. In the absence of monetary transactions or specifically targeted user functions (e.g. calendars), users can be reluctant to register and login (Bishop, 1998).

These characteristics require an approach which relies less on the registration of stable user profiles as is common in CF research, and more on the use of aggregated, short-lived, anonymous access patterns. To address these issues, we combined two frameworks, namely document relationship mining from DL logs and Spreading Activation (SA) recommendations to produce video recommendations for a multimedia DL. This methodology comprises three stages. First, DL logs and metadata are harvested by means of manual or automated transfer. The DL logs record anonymous user document downloads that occurred within a given period of time. Metadata for that set of documents is harvested via the OV Project's OAI-PMH (Van de Sompel et al., 2002) interface. Second, an analysis module derives document co-downloads from the pair-wise sequences of video downloads and uses these to generate a video relationship matrix. Third, a recommender module issues SA recommendations using the generated document relationship matrix.

## 2. Log analysis

The proposed method of log analysis is geared toward the derivation of document relationships from temporally ordered pairs of downloads. As such it deviates from traditional association rule data mining which is geared towards the detection of item associations from "baskets" of user purchases. We will briefly discuss the particular algorithm we applied to the generation of video relationships from the anonymous Open Video download logs.

### 2.1. General principles

User document downloads do not necessarily indicate a stable, continued interest which can be used to build a reliable user profile. For example, online holiday shopping for your teenage nieces and nephews can inject "interests" into your profile that you might not enjoy year-round. Furthermore, as discussed, recording such profiles is practically and legally impossible for many DLs.

Therefore, we make the minimal assumption that two documents downloaded one after the other, in close temporal proximity, and by the same, anonymous user, are more likely to be semantically related than not, since their downloads most probably originated in the same user information need. For example, a user may be looking for a document on the subject of CF recommendation systems and download another document related to this subject 5 min later. The fact that these particular documents were downloaded by the same user, in that specific sequence is the result of their common relationship to the same, albeit temporary, user information need. Therefore when we observe that a pair of documents are downloaded by the same user, one after the other and in close temporal proximity we label such an event a co-download which provides support for the assumption that the documents involved are thematically or conceptually related.

A single co-download can be a mistake or coincidence, but the frequent co-download of two documents in a specific sequence provides a much stronger indication that the documents are indeed related. For example, photos of a Squarepusher concert may be downloaded repeatedly shortly before an Autechre video clip is downloaded because a community of electronic music fans tends to think the two bands are strongly related. Overlapping patterns of document co-downloads over a community of users can therefore be used to gradually update sets of document relationships.

Defined as such this procedure resolves many of the mentioned constraints in which DLs operate with regards to the registration of user profiles:

1. It does not require information on the particular user's identity, merely the fact that the downloads were associated to the same, anonymous user.
2. No long-term records of user downloads need to be maintained. Document relationships can be updated in real-time as a new pair of co-downloads occurs. Records can be destroyed immediately afterwards.

We note that this procedure causes document relationships to be temporally ordered, i.e. depending on the order of co-downloads a relationship between A and B requires the absence nor presence of a relationship between B and A. The temporality of accesses is thus taken into account.

### 2.2. Formal specification and complexity analysis

We represent a document network as the weighted directed graph $G = (V, E, W)$ where $V = \{v_1, v_2, \ldots, v_n\}$ denotes the set of $n$ documents $v_i$, $E \subseteq V^2$, and $W$ is a weight function which associates each pair of documents $(v_i, v_j) \in E$ with a relationship weight so that $W(v_i, v_j) \rightarrow \mathbb{R}^+$. We represent the graph $G$ by the $n \times n$ matrix $M$ whose entries $m_{ij} = W(v_i, v_j)$.

We then denote a co-download of two documents $v_i$ and $v_j$ as the triplet $c(v_i, v_j, t(v_i, v_j))$ where $t(v_i, v_j)$ is the time passed between the retrieval of document $v_i$ and $v_j$ also referred to as the download latency. A co-download is observed on the condition that $v_i$ and $v_j$ have been downloaded by the same user and that its download latency is below a given threshold, i.e. $t(v_i, v_j) < \Delta_t$. The download latency can be used to apply a threshold to the determination of which co-downloads are taken into account, or how they are weighted in terms of generating document relationships. Given that a set of $k$ co-downloads $C = \{c_1, c_2, \ldots, c_k\}$ has been extracted from a DL log, we can generate matrix $M$ by defining each entry $m_{ij}$ as a function of the frequency of co-download of the documents $v_i$ and $v_j$ as follows.

**set each** $m_{ij} = 0$
**foreach** $c_k = (v_i, v_j, t(v_i, v_j)) \in C$
{
  **if** $t(v_i, v_j) < \Delta_t$
  **then** $m_{ij} += f(c_k)$
}

$f(c_k)$ represents the edge weight reinforcement function which can be varied based on the properties of the co-download event, e.g. the length of the download latency, as discussed in Section 4.2.

The items in a DL download log are temporally ordered since they are recorded in the sequence that they occur. The outlined procedure thus consists of linearly scanning the downloads in a DL log, determining the time elapsed between any two subsequent downloads, decide whether they correspond to the same user, and insert or update the appropriate document relationship edge.

The complexity of this algorithm is therefore given by $O(n)$ where $n$ represents the number of recorded downloads in the log. As such the computational costs of the algorithm will scale linearly with the size of the log data. Furthermore, such a linear scan can be conveniently broken down into multiple sub-tasks to be executed in parallel by splitting the logs in $n$ sections.

Since co-downloads will overlap, we expect the density of the resulting document matrix, denoted $d$, to be $d \ll n$, resulting in highly sparse matrices which can be efficiently stored using Compressed Column Storage (CCS) formats such as discussed in Duff et al. (1989). Such formats only store the non-zero entries of a matrix of which there are much less than the possible $n \times n$ entries of a dense matrix.

### 2.3. Related research

The outlined principle of using pair-wise sequences of downloads is similar to clickstream analysis which derives association rules from usage data (Chan, 1999; Ahmad Wasfi, 1999; Mobasher et al., 2001). This problem is usually framed in data mining research as the discovery of significant associations between items on the basis of their co-occurrence in consumer purchase "baskets" (Agrawal et al., 1993), also known as market basket analysis. For example, if "beer" and "diapers" often co-occur in sets of user purchases, i.e. the "baskets", we may derive the rule that purchasing "beer" leads to purchasing "diapers with a 0.75 probability and a support of 100 observations. In this example, "beer" would be labeled the antecedent and "diapers" the consequent of the generated association rule.

Our co-download definition deviates from that approach by only considering temporally ordered pairs of video downloads, i.e. a user basket can only contain two items which are furthermore required to occur in close or immediate temporal proximity. As such the space of possible antecedent-consequent rules is severely restricted resulting in a low computational costs of the algorithm, i.e. $O(n)$. This method has for that reason been applied to large-scale DLs (Nelson et al., 2004) and DL journal linking (Bollen and Luce, 2002).

### 2.4. Characteristics and limitations

Since it relies on usage data the outlined procedure for the generation of document relationships from ordered download pairs will be subject to a number of problems:

1. Consistent user biases: user behavior is determined by a multitude of factors, such as user interface features, existing document link services, document metadata, etc. If such factors consistently bias specific download behaviors, e.g. documents with salient but misrepresentative titles, they will affect download data and cause erroneous document relationships to be generated.
2. Lack of sufficient data: each sequential download of two documents provides further support to the assumption that they are related. However, a lack of download data will lead to weakly weighted or absent document relationships. Absence of download data, however, does not equate the absence of document relationship. This is the "boot-strapping" or "cold-start" problem: until the system reaches steady state, spurious relationships will be hard to distinguish from relevant relationships.
3. Majority rule: although each download counts, if a majority of users persistently follows a particular download sequence, the majority view will be most strongly

valued in the resulting document networks. Although minority behaviors will be reflected in the generated document relationships, they may be overshadowed by majority preferences and biases.

4. Use of IP addresses: due to the use of proxies there is no guarantee that two requests originating from the same IP address actually correspond to the same user. Unfortunately, due to the limitations mentioned in Section 1, the IP address is often the only available means of tracing user requests in a DL setting. However, as long as the pair-wise sequential order of requests are maintained the problem of falsely reinforcing edge weights in the document graph is limited to instances where multiple user requests temporally overlap for the same IP. Additional techniques, such as combining the IP address with the HTTP user agent field (which is frequently unique), can be used to lessen the chances of IP address collision.
5. Robot accesses: if a web robot crawls a web site and sequentially downloads all resources, then non-semantic relationships can be reinforced. Fortunately, there are a number of web engineering methodologies (e.g., "page tags") that can be used to identify and eliminate accesses by non-interactive users.

The above listed problems may reduce the validity of generated document relationships, but the extent to which this is the case is an empirical matter. It should be determined from an actual analysis of the performance of any systems that rely on download data, as we have attempted in this report.

## 3. Spreading activation recommendations

Document recommendations can be extracted from the document matrix $M$, but it is expected to be highly sparse: the number of retrievals will be well below the possible number of document relationships. This problem has been addressed by Huang et al. (2004) who demonstrated that associative retrieval techniques can expand a small initial recommendation set resulting from matrix sparseness.

We have employed a similar result expansion technique known as Spreading Activation (SA). Although SA has originally been formulated as a model of associative retrieval from human memory (Anderson, 1983; Collins and Loftus, 1975), it has found ample applications in IR systems (Cohen and Kjeldsen, 1987; Crestani, 1997; Crestani and Lee, 2000). We have successfully constructed SA recommender systems on the basis of document and journal networks generated from DL logs (Bollen, 2000, 2001) and have therefore chosen to address possible document relationship sparseness in this manner.

The general process of SA can be described as follows. We assign an initial activation value $a_i$ to every document $v_i$. The set of activated documents represents our query. Activation values are then propagated from that set of initially activated documents through the network of document relationships. Documents which accumulate most activation after a particular number of iterations are marked to be retrieval results. SA thus simulates a process whereby an initial set of documents is

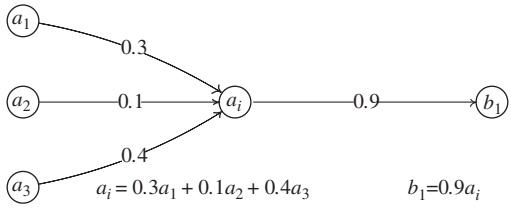


Fig. 1. The activation value of a node is given by the sum of the in-linking nodes' activation values.

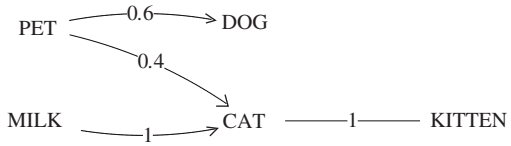


Fig. 2. Example network.

expanded by passing activation values to all directly and indirectly connected documents.

The propagation of activation values is defined such that the activation value of each node $a_i$ is determined as a function of the weighted sum of the activation values of all documents that link to it, or $a_i = f(\sum_{\forall_j} m_{ji}a_j)$, as shown in Fig. 1. The activation function $f$ represents a specific model of how the node retains or leaks activation between iterations, e.g. a leaky capacitor model (Anderson and Pirolli, 1984).

As an example of this procedure, assume we have created a set of document relationships as shown in Fig. 2.

We query this document network by activating the nodes "MILK" and "PET", i.e. their activation energy is set to 1. Subsequently these activation values spread to the adjacent documents via the network connections. The activation energy received by each node is calculated as the sum of the weighted activation values it receives from its antecedent nodes:

```
PET = 1
MILK = 1
DOG = PET * 0.6 = 0.6
CAT = PET * 0.4 + MILK * 1 = 1.4
```

For simplicity, we assume no decay is applied to the resulting activation values. The next iteration of the SA algorithm will further propagate these activation values:

```
KITTEN = CAT *1 = 1.4 * 1 = 1.4
```

After a given number of iterations, we rank the nodes according to the activation values they have received. In this case, "CAT" and "KITTEN" followed by "DOG" would be our top-ranked results for the query "PET & MILK".

Given we have a matrix of document relationships $M$, we can simulate SA by a process of repeated matrix-vector multiplications. To this end, we represent the activation state of the network at time $t$ by the vector $a_t$ such that each entry of the activation vector $a_t(i)$ corresponds to the activation value of a network node $v_i$. We denote the initial activation vector $a_0$. Each subsequent activation state is then given by

$$a_{t+1} = (f(I) + M) \times a_t.$$

The entries of the identity matrix $I$ represent the activation function $f$ which may apply a threshold function to activation values or reduce each activation value by a certain factor $\lambda < 1$ between iterations (leaky capacitor model). We repeat this process by $k$ iterations so that the values of $a_{t=k}$ represent the network's final activation state. The nodes of the network can then be ranked according to the entries of the activation vector.

The process of SA is attractive for IR applications since it establishes the relevance of a document for a given query according to the overall structure of document relationships which can be defined independent of document content. Since activation spreads in parallel through the network, it can find pathways between related documents which could not have been identified by term matching or other procedures. Due to its parallel nature it is furthermore resistant to minor errors in network structure and sparseness of association data. However, SA requires the generation of extensive document networks (Woodruff et al., 2000) which has proven to be a considerable hurdle to its general application. Given the above mentioned methodology for the generation of document relationship networks from DL logs, we find SA an efficient and promising recommender technique for DLs.

## 4. Open Video Project: a test case

Our objective was to demonstrate we can rapidly prototype SA recommender services for multimedia DLs using the above described log analysis and SA techniques. We chose the Open Video Project (Slaughter et al., 2000) because of its focus on multimedia documents and the fact that it is OAI-PMH compliant, i.e. it is instrumented to accept OAI-PMH requests for metadata (Van de Sompel et al., 2002).

The Open Video Project began as an effort to create a large collection of varied video content that could be used as a testbed for research in areas such as human–computer interaction, multimedia IR, and technical video processing techniques such as automated video segmentation (Geisler et al., 2001; Wildemuth et al., 2003). In addition to researchers, its user community now includes multimedia artists, students and instructors, as all members of the general public. The collection currently contains over 2000 unique video segments, available in

MPEG-1, MPEG-2, and MPEG-4 formats. Open Video exposes its metadata in Dublin Core through the OAI-PMH.

### 4.1. Open Video Project logs

The Open Video Project records standard web site usage data in its server logs, including pages viewed, search terms used, and video segments downloaded. These user actions are sorted in the logs by the date and time they were recorded. We only considered user document download requests for the reconstruction of co-download events. The Open Video log files register download date and time, user IP address, the video identifier (URL pointing to storage location) and a segment identification number for such requests. We analyzed log data registered between 07/10/2002 and 11/04/2002, containing 10 954 retrieval requests for 1320 videos over a total of 1841 available videos. We harvested the metadata pertaining to all 1841 video segments using the OAI-PMH from which we retrieved video titles and IDs.

To ensure the privacy of individual users we substituted IP addresses with unique user identification numbers. Although using the IP address to identify users is prone to errors given the frequent use of proxy servers and the fact that many users share the same computer and IP address, few alternatives are available to associate document downloads to specific users in many, if not most, DLs. Our results indicate this shortcoming does not hamper the production of viable document networks. Possible future solutions to this problem may involve removing proxy IP addresses from a DLs logs before processing or reducing the influence of highly frequent IP addresses.

### 4.2. Estimation of log analysis parameters

The proposed log analysis method requires the determination of an appropriate value of the $\Delta_t$ parameter, the value which is used to determine whether two subsequent download correspond to a co-download event. A $\Delta_t$ which is high will increase the odds of false positives, i.e. the number of download pairs which are erroneously considered a co-download event. Conversely, a low $\Delta_t$ will be more stringent on which pairs of downloads to consider a co-download, but increase the probability of false negatives, i.e. pairs of downloads which do represent valid co-downloads will be rejected. The issue is thus to determine a value of $\Delta_t$ which minimizes both false positives and negatives.

Rather than to apply a binary $\Delta_t$ threshold, we attempted to produce a function that estimated the probability that a pair of downloads constituted a valid co-download. The central assumption of this approach is that for any two video downloads to be related according to their download sequence a user ideally had to have at least fully downloaded and seen them. The expected combined delays incurred by downloading and watching a video will determine the value that we expect the latency between video downloads to be given ideal circumstances. When download latencies deviate from this ideal, they will not be discounted but assigned a lesser weight: they may concern a user or robot who quickly downloads one video

after the other without regard for their content, or a user who interrupts a download session and returns to continue with another one.

Since we assumed download latencies to be shaped by two factors, i.e. video download times and video durations, we had to produce estimates for both and combine them. To determine expected video duration we randomly selected 20 videos from the log file and determined their mean duration in seconds which proved to be 709 s. To determine expected video download times we then selected a set of 20 video with a duration of approximately the average duration of 709 s. These videos were downloaded via the university's connection and average download times were determined to be 247.05 s. Combining the distributions of movie duration and average download times, we obtained a distribution which corresponded to the expected download latencies between videos if a user fully downloaded and watched each video before downloading the next one. The mean of this distribution was 1015 s with a standard deviation of 451 s.

A frequency distribution was constructed to determine how many videos in our test set would match the expected duration and download times. This distribution was fitted with a Gaussian function to produce the model of expected download and duration times as shown in Fig. 3.

The log analysis can then proceed on the basis of this model. For each pair of video downloads in the logs, the actual download latency would be matched against the generated Gaussian model. A download matching the exact mean of the distribution would be maximally reinforced i.e. the reinforcement is 1. All other downloads would be reinforced according to the degree to which their

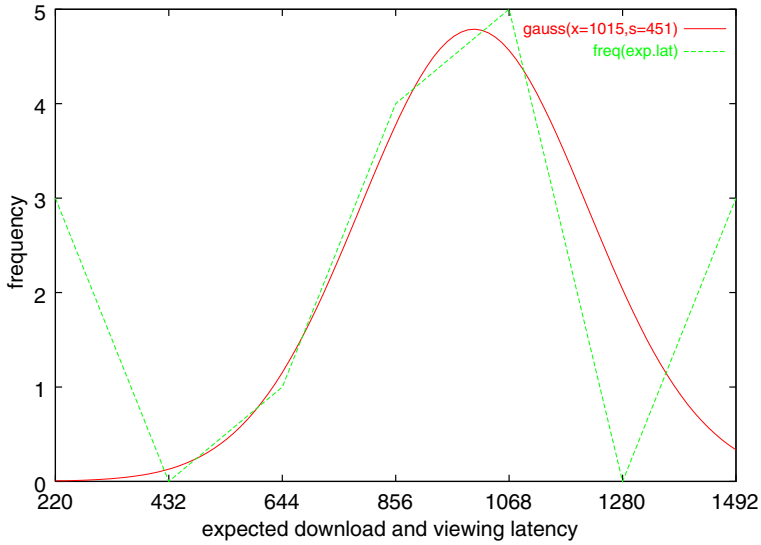


Fig. 3. Download and viewing latency fit.

download latencies positively or negatively deviated from the Gaussian model's mean.

We acknowledge this procedure can only provide a rough estimate for a desirable value of $\Delta_t$ and more statistically advanced mechanisms need to be explored. However, the precision and recall analysis discussed in later sections confirms it yields valid recommendations.

### 4.3. A network of video relationships

From the harvested log data we generated a network of weighted video relationships represented by a $1841 \times 1841$ matrix according to the methodology described in Section 2.1. Co-downloads were reconstructed for all pairs of video downloads by the same user within the same day. The resulting video network was exceedingly sparse; only 7147 video relationship were created. Fig. 4 shows two sub-networks of videos related to "natural disasters" and "love". As indicated by these graphs the generated network of video relationships was sparse compared to the range of possible relationships but provided a rich set of connections among significant sets of videos. In Section 5.1 we provide an overview of the recommendations that can be derived from these video relationship networks.

A Java application was implemented for the automated analysis of DL logs. The application program can read any DL log file which contains date and time stamp, user identification number, and document identifier. At present we convert the Open Video log to an internal format, but future versions may apply proposed standards (Goncalves et al., 2002) or (Van de Sompel et al., 2003).

The generated matrices were very sparse and therefore stored in the mentioned CCS format which is similar to the Harwell-Boeing standard (Duff et al., 1989). The CCS format stores a matrix as three arrays which contain column pointers, row indices, and non-zero matrix values. By not storing zero valued entries, the CCS format can efficiently store large, sparse matrices depending on the number of non-zero matrix entries.

CCS furthermore allows sparse matrix-vector multiplications to be performed in $2nnz$ operations where $nnz$ represents the number non-zero matrix entries. Efficient matrix-vector applications are required for the generation of SA recommendations as discussed in Section 3.

### 4.4. Open Video spreading activation recommender service

A SA recommender system, as discussed, was implemented to operate on the generated document relationship matrix. The present architecture consists of two main architectural components, namely the log analysis module discussed above and the SA module, which internally communicate to assure log and metadata information are continuously up to date. The objective of this architecture is to efficiently perform both log analysis and prototyping of SA recommender system within the same architectural framework.

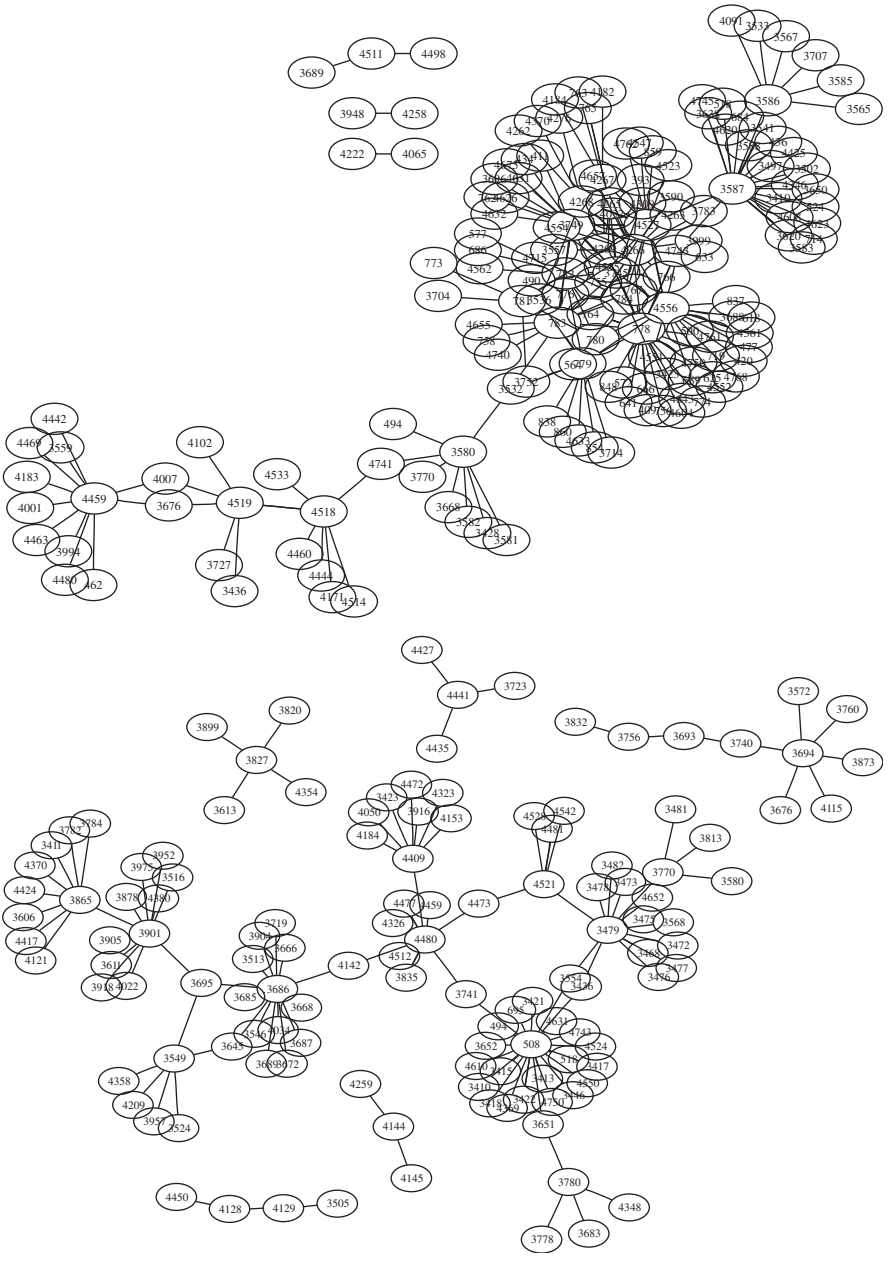

Fig. 4. Sub-networks related to "natural disasters" (top) and "love" (bottom).

### 4.4.1. Recommender module architecture

The recommender module (Fig. 5) was implemented in the form of a publicly accessible Java Servlet which consisted of three modules:

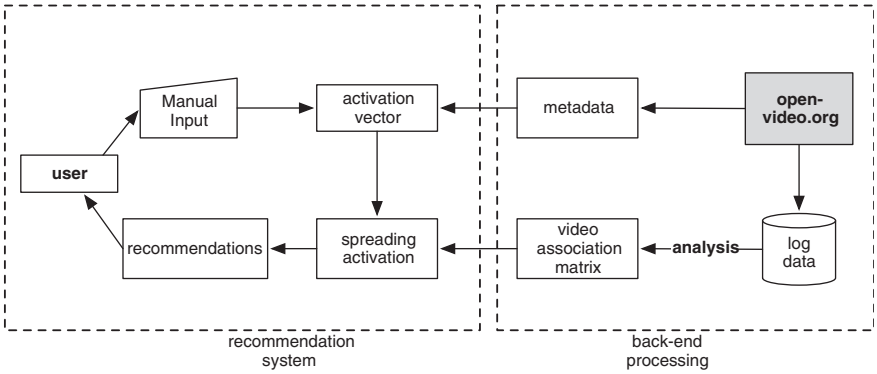


Fig. 5. Process flow in the spreading activation recommender system.

*Query term matching module*. This module links query terms to video titles by matching the provided query terms to a stored list of video titles, or any other metadata available. Document activation levels correspond to the number of matches for a specific video title normalized by term inverse document frequency (IDF).

*Spreading activation module*. Implements the actual SA algorithm. The algorithm carries out an iterative multiplication of the row-normalized association matrix and the generated initial activation vector. For each document a maximum activation value over all iterations is calculated. Documents with high activation values are marked as potential recommendations.

*Interface*. A servlet implements the user interface. It receives user queries in the form of document identifies to allow third party searching and term queries to allow human searches. Recommendations are ranked according to their activation values on the basis of an initial set of documents activated by the document ID or term queries.

### 4.4.2. *Query processing*

The recommender service was originally designed to respond to query requests consisting of sets of document IDs for which recommendations needed to be issued. This approach would in fact void the need for metadata harvesting since a query would be represented by no more than a set of document identifiers. However, in our prototypes such a procedure would hamper our ability to explore the effectiveness of the system since a tester would need to be aware of all document IDs to issue effective queries. For this reason we adopted an initial querying by keyterms procedure which assigned initial activation values to videos according to how well their titles matched the query terms.

The relation between query term matches and initial video activation values was determined on the basis of a term weighting scheme known as Term Frequency vs. Inverse Document Frequency (TFIDF) (Salton, 1989, 1998). TFIDF weighting

balances two factors in determining the importance of a video title-term match, namely Term Frequency (TF) and Document Frequency (DF). First, the frequency with which a term matches a video title, i.e. its TF, indicates the degree to which it is a good representation of the video's content. Second, some terms are simply more frequent than others and more likely to match any given video title. Therefore, the TF of a term needs to be balanced by its overall frequency in the collection, i.e. its DF. The final term weight is thus a function of the ratio of the terms' TF and its DF. The activation of each video was then determined by the sum of the TFIDF weights of all terms that matched the particular video's title.

Since video titles are short, one-sentence descriptions of a video's content, repeated matches of the same term will not be meaningful. We therefore assume that the TF of each term will be 1. All video title terms were previously separated and stemmed using the Porter algorithm (Porter, 1980). Given that each term's TF was 1, the resulting TFIDF weight of a term $q$ matching a video $i$, labeled $w_{i,q}$, was then defined as

$$w_{i,q} = \log\left(\frac{N}{n_q}\right),$$

where $N$ represents the total number of documents in the collection, and $n_q$ represents the number of videos whose titles contained the term $q$ (DF). $n_q \geqslant 1$ since terms have been derived from video titles and each term must therefore match at least one video title.

The initial video activation values are then defined as the sum of the TFIDF weights of all query terms that the particular video title matches:

$$a_{t=0}(i) = \sum_{\forall q \in i} w_{i,q}.$$

This procedure ensures that highly common query terms do not simply activate all documents in the collection; because of their high DF values, their TFIDF weights will be low and they will exert little influence on activation values. Low DF query terms will on the other hand exert a relatively high impact on a document's initial activation value.

## 5. Evaluation

We evaluated the effectiveness of the process used to derive video recommendations from DL download logs using two methods. First, we subjectively examined the generated recommendations for a set of queries which would have yielded few useful results on the basis of the existing OV web site's text search. Second, we objectively compared its performance to a CF system operating on the same log data. Our definition of performance is based on the assumption that both systems would be employed to generate a set of recommended videos for a given query video, i.e. a user indicates an item of interest and the system responds by returning a set of recommendations, i.e. the "find good items" task (Herlocker et al., 2004). The

performance of both systems is expressed in terms of precision and recall values for the relationships they generate for a set of the 10 most downloaded videos. An analysis indicates the log analysis method outperforms a CF system for the same data set.

### 5.1. Qualitative evaluation

As a preliminary assessment of the potential of our methodology to effectively help users find video documents related to a particular information need, we have created a simple web-based interface that displays recommended video titles for a submitted query. In this interface, results for a submitted query are displayed in two sections: ''term matches'' displays video documents whose titles match the submitted query terms, while ''recommendations'' displays video documents that are recommended by the SA process.

A large section of the OV collection is dedicated to post World War II educational movies. We therefore first used the query term ''fallout'' to investigate the topic of nuclear weapons. The existing OV text search function produces only two video titles based on term matches, but the SA system is able to produce numerous recommendations, many of which are highly relevant. For example, videos such as ''Duck and Cover,'' ''Stay Safe, Stay Strong: The Facts About Nuclear Weapons,'' and ''Atom Bomb'' are among the top five recommendations (Table 1). Note how the social interpretation of the term ''Fallout'' is represented in a set of recommendations relating to matters of popularity and family life.

Similarly, we issued a query for the term ''Moon'' in both the OV web site's term match and our SA recommendation system. The results are listed in Table 2. Again we find the term match produces exactly 1 relevant result which carries the term ''Moon'' in its title. The SA system generated more than 30 recommendations of which the 10 highest ranked strongly focus on the Apollo program and exploration of the ocean floor.

Table 1
Spreading activation recommendations for ''Fallout'' query compared to text matches

| Rank | Term match | SA recommendations |
|---|---|---|
| 1 | About fallout (1955) | Duck and cover |
| 2 | About fallout (1963) | According to plan: the story of modern sidewalls for the homes of America |
| 3 | | Stay safe, stay strong: the facts about nuclear weapons |
| 4 | | Atom Bomb [Joe Bonica's movie of the month] |
| 5 | | Are you popular? |
| 6 | | A is for Atom |
| 7 | | Medical aspects of nuclear radiation |
| 8 | | As boys grow |
| 9 | | Are you ready for marriage |
| 10 | | Angry Boy (Part I) |

Table 2
Spreading activation recommendations for "Moon" query compared to text matches

| Rank | Term match | SA recommendations |
|---|---|---|
| 1 | Moon, segments 01-08 | Apollo, segment 1003 |
| 2 | | Apollo, segment 3001 |
| 3 | | Story rooms |
| 4 | | Oceanfloor legacy, segment 01 of 14 |
| 5 | | Apollo segment 6012 |
| 6 | | Oceanfloor legacy, segment 10 of 14 |
| 7 | | Apollo, segment 3002 |
| 8 | | Airport, The |
| 9 | | Apollo, segment 4002 |
| 10 | | Wood for war |

Although the mentioned results indicate the ability of the system to produce a larger number of more relevant results, a more objective evaluation of the quality of the SA generated recommendation was required. The following section will compare the precision and recall of the SA recommendations with recommendations generated by means of a CF approach.

### 5.2. Comparison to collaborative filtering

Most videos in the Open Video archive lack significant text metadata, other than abbreviated titles and a video identifier. An example is shown in Fig. 6: the video entitled "NASA 25th Anniversary Show" which features NASA's space shuttle does not contain the terms "Space shuttle" in its title nor video ID. In addition, no description or abstract is available. A query for the terms "space shuttle" would not match this video.

To determine the efficacy of our own recommendation methodology by a comparison to other established methods, we thus had to focus on methods that do not rely on text or metadata. In this case we generated an alternative set of recommendations based on a traditional CF model, namely item-based CF (Sarwar et al., 2001).

To generate a set of CF recommendations for each video, we first create a set of user profiles from the available download data. Each user profile consists of the videos downloaded by that user during the period in which the download logs were registered. This procedure corresponds to the behavior of a service like *amazon.com* which registers a user's purchases over time and stores it as their profile. The set of all user profiles defines a video-user adjacency matrix $U$ whose entries represent how often a video has been downloaded by a particular user. In this matrix $U$, each video is thus represented by a row vector of download frequencies. Similarly, each user can be represented by a column vector of matrix $U$.

We use TFIDF weighting to determine the relationship weight between users and videos. These weights will indicate how well a particular user is associated with a

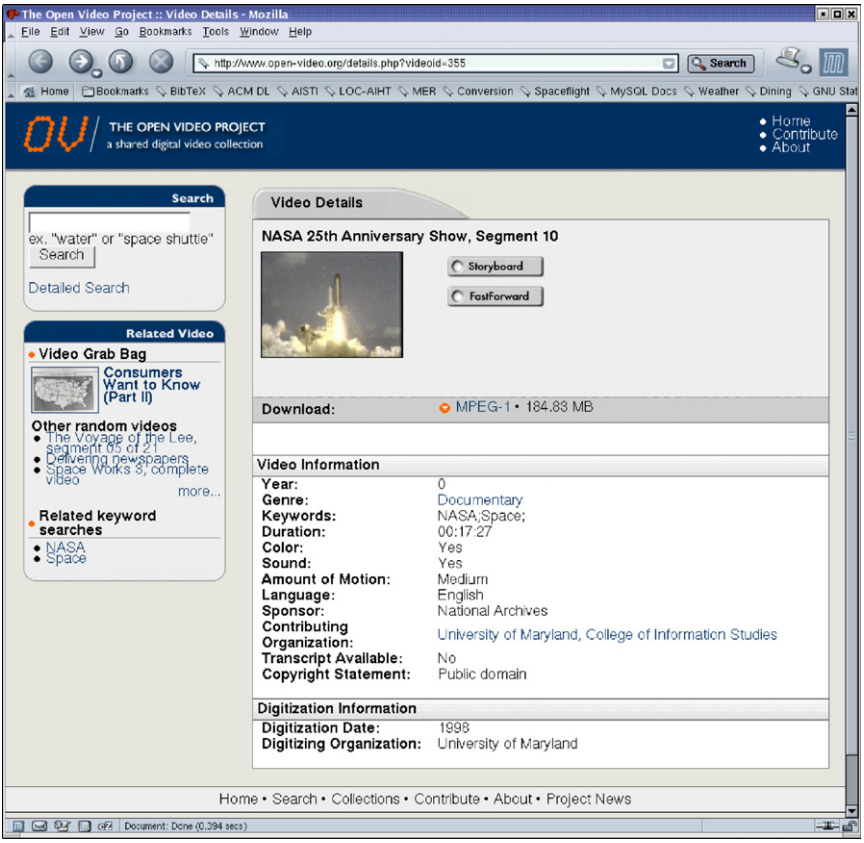


Fig. 6. Metadata for movie about NASA's space shuttle lacks terms referring to space shuttle.

particular video. Clearly, a user that has downloaded many or all videos, e.g. a spider or robot, will not be strongly attached to an individual video. On the other hand, a user that has downloaded only two videos will be highly specifically tied to that video in terms of our objective to represent a video by a vector of user weights. We apply a TFIDF weight so that

$$w(v_i, u_j) = \frac{\mathrm{TF}}{\mathrm{DF}},$$

where

$$\mathrm{TF} = \frac{\mathrm{freq}_{i,j}}{\max_l \mathrm{freq}_{l,j}} \quad \text{and} \quad \mathrm{DF} = \log\left(\frac{N}{n_i}\right).$$

In this case, $\mathrm{freq}_{i,j}$ represents the frequency by which a video $j$ was downloaded by a user $i$ and $\max_l \mathrm{freq}_{l,j}$ the maximum number of times the user has downloaded a particular video as a normalization. The DF is determined by $N$ which represents the total number of videos and $n_i$ which represents the number of videos downloaded by

the particular user *i*. In other words, the weight with which we associate a particular user to a video is determined by the frequency by which the user downloaded that particular video normalized by the total number of videos downloaded by that user. In this manner robots and spiders, who tend to indiscriminately download large numbers of videos, will have a weak association to each individual video (high DF), while genuine users will, depending on which videos they downloaded, be significantly associated to a set of particular videos (TF high relative to DF).

All quantities needed to determine the TFIDF weight between a video and a user can be retrieved from matrix $U$. For each pair of users and videos we can thus determine a TFIDF weight thereby defining a TFIDF video-user matrix which we label $U'$. In $U'$ each video is represented as a vector of TFIDF user weights, i.e. a video is characterized by the set of users who downloaded it. We assume that videos downloaded by similar sets of users are semantically similar. The latter follows from a basic assumption of CF models: since users have stable tastes and can be expected to download documents to match these preferences, documents downloaded by the same users tend to be similar. To calculate video similarity from the generated video-user TFIDF weights we then define the degree of relationship of each pair of videos $S(v_i, v_j)$ as the cosine of the angle $\alpha$ of their user vectors:

$$S(v_i, v_j) = \cos(\alpha) = \frac{\vec{v_i} \cdot \vec{v_j}}{\|\vec{v_i}\| \times \|\vec{v_j}\|}$$

or the normalized inner products of the user vectors of the two videos.

For each pair of videos we can thus calculate a cosine similarity value, and thereby for each video generate a set of video recommendations ranked according to their cosine similarity. These recommendations can be compared to those issued by the earlier discussed log analysis and SA method.

### 5.3. Identification of relevant recommendations

A set of the 10 most frequently downloaded videos was derived from the OV download logs as listed in Table 3. For each of these we identified and ranked a set of relevant recommendations, i.e. those videos that would be considered most related in terms of content and general features to the specific video. The process by which we determined an ideal set of relevant recommendations for each video proceeded according to four discrete phases:

1. identification of video content;
2. manual scanning of all video titles to identify potentially relevant recommendations;
3. pruning of initial set of recommendations by analysis of content and ranking of relevance to particular video;
4. group evaluation of final set.

We thus generated a set of relevant videos for each of the 10 most downloaded videos. These sets may not have been complete or entirely logically consistent. The

Table 3
List of 10 most frequently downloaded videos

| Video ID | Video name | Description |
|---|---|---|
| 4743 | 2 a.m. in the subway | Drama, theater |
| 3532 | Atom Bomb Joe Bonica's Movie of the Month | Atomic weapons testing |
| 3415 | A is for Atom | Educational - Nuclear Physics |
| 431 | Apollo, Segment 1001 | Apollo Moon landing |
| 4531 | The Your Name Here Story | Industrial Film |
| 3641 | Chinese Lion Dance: Marysville, California143 | Bok Kai Festival, CA |
| 4737 | A ballroom tragedy | Drama, theater |
| 3741 | Duck and Cover | Classic 50's educational |
| 3651 | Classic Television Commercials (Part I) | Collection of TV commercials |
| 4590 | "Columbia" winning the cup | 1899 Yacht Race |

identification of relevant recommendations remains a subjective matter. However, we attempted to identify those videos which were most likely to be considered functionally adequate recommendations given a particular video, regardless of their title and other superficial identifications. Table 4 lists a number of videos that were selected as relevant recommendations for the video "Duck and Cover". In this case our selection could not be guided by the presence of keyterms; the terms "Duck and Cover", do not adequately indicate the meaning and content of this video. Rather we interpreted its content as that of an educational video focused on surviving nuclear disaster. Our selection of relevant videos was then guided by this interpretation. The selection of the video "Survival Under Atomic Attack" was clearly the result of this process, as well as "About Fallout (1955)" which although it is described as an attempt to dispel myths about nuclear fallout still matches the requirements for an educational movie about the effects of a nuclear disaster. Although our selection could not possibly be complete we attempted to at least identify the core set of relevant items and verify their relevancy with the OV project design and implementation team.

### 5.4. Precision and recall

We calculated the precision and recall for recommendations issued for the 10 most downloaded videos by the log analysis and SA method discussed in Section 2, labeled SA, and the CF method discussed above, labeled CF. Precision was defined as the ratio of the number of relevant items among the SA and CF answer sets over the number of total number of items in the answer set (10). Recall was defined as the ratio of relevant items in the answer set over the total number of relevant items identified for a particular video.

The results of our experiments are summarized in Table 5. As can be seen, the SA algorithm performs better than the CF algorithm for both precision and recall in virtually all cases. The SA algorithm outperforms the CF for 8 out of 10 videos in terms of its precision. The SA algorithm had an average precision of 17.51%, while

Table 4
List of videos considered relevant recommendations for "Duck and Cover" video

| Duck and cover | |
|---|---|
| Title | Description |
| Survival Under Atomic Attack | Educational - Atomic Attack survival strategies |
| Atomic Alert (Elementary version) | Education - Attack survival tips |
| News Magazine of the Screen, The | News, atomic attack disaster |
| News Magazine of the Screen, The (5,10; 1955) | News, disaster |
| About Fallout (1955) | Educational, fallout |

Table 5
Precision and recall results of recommendations issued by two methods (SA and CF) for 10 most frequently downloaded videos

| Video ID | Video name | Precision | | Recall | |
|---|---|---|---|---|---|
| | | CF (%) | SA (%) | CF (%) | SA (%) |
| 4743 | 2 a.m. in the subway | 0 | 11.67 | 0 | 63.63 |
| 3532 | Atom Bomb Joe Bonica's Movie... | 0 | 16 | 0 | 50 |
| 3415 | A is for Atom | 0 | 4 | 0 | 28.57 |
| 431 | Apollo, Segment 1001 | 83.33 | 46.43 | 22.72 | 29.54 |
| 4531 | The Your Name Here Story | 0 | 3.12 | 0 | 12.5 |
| 3641 | Chinese Lion Dance: Marysville, CA | 0 | 33.33 | 0 | 16.67 |
| 4737 | A ballroom tragedy | 0 | 17.24 | 0 | 62.5 |
| 3741 | Duck and Cover | 0 | 10 | 0 | 40 |
| 3651 | Classic Television Commercials (p1) | 0 | 33.33 | 0 | 50 |
| 4590 | "Columbia" winning the cup | 0.98 | 0 | 20 | 0 |

the CF algorithm had an average precision of 8.43%. This level of precision for a set of 10 recommendations is relatively low for both systems. Similar results are found in terms of recall. The SA algorithm outperformed the CF algorithm for 9 out of 10 videos in terms of recall. The SA algorithm had an average recall of 35.34% and the CF algorithm had an average recall of 4.27%.

We performed a Kolmogorov–Smirnov test over the precision and recall values of the CF and SA recommendations for all 10 videos to determine whether the higher level of performance of the SA methods was statistically significant. The null-hypothesis for the precision evaluation of CF and SA methods was rejected at $p<0.05$, while the null-hypothesis for the recall evaluation of CF and SA methods was rejected at $p<0.005$. In both cases we can thus confirm that the SA methods significantly outperforms the CF method.

As expected the recall numbers of the SA algorithm over 10 recommendations exceeds its precision. Due to SAs behavior of conducting an associative, parallel

search through a network of document relationships, SA will generally succeed in identifying a wide range of direct and indirect paths to possibly related documents. This will increase its recall but hurt its precision. Huang et al. (2004) discuss the problem of sparsity in item-based recommender systems and how it can be alleviated by the use of associative retrieval techniques. Their conclusion is born out with these results: the use of item-based recommendations is hampered by the sparsity of usage data as compared to the number of available items. Item-based CF techniques will thus in many cases fail to produce any recommendations as evidenced by the zero valued precision and recall numbers in Table 5. This problem can be alleviated by the use of associative retrieval techniques such as SA. A comparison of CF approaches to the minimal log analysis method in this paper, but using equal recommendation generation techniques, is thus warranted.

## 6. Conclusions

We have developed a system which can rapidly generate recommender services for multimedia DLs, provided they record at least two subsequent user document downloads. The generated recommender systems can be integrated into existing user interfaces as a third party recommender service and be engaged in an automated feedback loop in which recommendations are continuously improved by additional user downloads. We demonstrated the viability of such a system for the Open Video Project, and have shown its ability to recommend video documents on the basis of semantic relationships established from past records of user downloads.

Similar recommender systems as those that have been discussed in this article have been developed for the Los Alamos National Laboratory Research Library (Rocha and Bollen, 2000) and the NASA Technical Report Service (NTRS) (Nelson et al., 2004). These results demonstrate the applicability of the log analysis methodology for DLs with a range of different media-formats. The reconstruction of co-download events is bound by a number of parameters which may need to be optimized for different collections and user interfaces.

Although the present spreading activation recommendation system has a number of promising characteristics, it need not be the sole recommender service implemented under this framework. Different recommendation systems, for example Bayesian models and belief networks, could be implemented to complement the existing spreading activation recommender module. Information retrieval techniques, such as Vector Space Models, could be used to seed an initial set of document association which could be further refined on the basis of log data. We plan to expand the present log analysis module with a feature that generates automated reports on the structure of user download preferences. The proposed service would thus not only produce recommender systems, but may find applications in the automated evaluation of DL collections (Bollen and Luce, 2002; Bollen et al., 2003) and user communities for a wide variety of document formats.

## Acknowledgments

This research has partially been supported by a grant from the Los Alamos National Laboratory Research Library. We extend our gratitude to Rick Luce and Herbert Van de Sompel who have contributed many of the ideas outlined in this paper. We also thank Somasekhar Vemulapalli and Weining Xu for their work on the systems presented here.

## References

Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: ACM SIGMOD international conference on management of data, Washington, DC; 1993. p. 207–16.

Ahmad Wasfi AM. Collecting user access patterns for building user profiles and collaborative filtering. In: Proceedings of the fourth international conference on intelligent user interfaces. New York: ACM Press; 1999. p. 57–64.

Anderson JR. A spreading activation theory of memory. J Verbal Leaning Verbal Behav 1983; 22:261–95.

Anderson J, Pirolli PL. Spread of activation. J Exp Psychol Learning, Memory Cognition 1984; 10:791–8.

Bishop AP, Logins and bailouts: measuring access, use, and success in digital libraries. J Electron Publishing 1998; 4(2) URL ⟨http://www.press.umich.edu/jep/04-02/bishop.html⟩

Bollen J. Group user models for personalized hyperlink recommendation. In: Lecture notes on computer science 1892—international conference on adaptive hypermedia and adaptive web-based systems (AH2000), Trento: Springer; 2000. p. 39–50.

Bollen J. A cognitive model of adaptive web design and navigation. PhD thesis, Vrije Universiteit Brussel, Brussels, Belgium; 2001.

Bollen J, Luce R. Evaluation of digital library impact and user communities by analysis of usage patterns. D-Lib Magazine 2002; 8(6) URL ⟨http://www.dlib.org/dlib/june02/bollen/06bollen.html⟩

Bollen J, Luce R, Vemulapalli S, Xu W. Detecting research trends in digital library readership. In: Proceedings of the seventh European conference on digital libraries, Lecture notes on computer science, vol. 2769. Trondheim, Norway: Springer; 2003. p. 24–8.

Chan PK. Constructing web user profiles: a non-invasive learning approach. In: Masand B, Spiliopoulou M, editors. Web usage analysis and user profiling—Lecture notes on artificial intelligence, vol. 1836. San Diego, CA: Springer; 1999. p. 39–55.

Cohen PR, Kjeldsen R. Information retrieval by constrained spreading activation in semantic networks. Inf Process Manage 1987;23(4):255–68.

Collins A, Loftus E. A spreading activation theory of semantic processing. Psychol Rev 1975;82:407–28.

Crestani F. Application of spreading activation techniques in information retrieval. Artif Intell Rev 1997;11(6):453–582.

Crestani F, Lee PL. Searching the web by constrained spreading activation. Inf Process Manage 2000;36(4):585–605.

Duff IS, Grimes RG, Lewis JG. Sparse matrix test problems. ACM Trans Math Software 1989;15(1): 1–14.

Geisler G, Marchionini G, Nelson M, Spinks R, Yang M. Interface concepts for the open video project. In: ASIST 2001: proceedings of the 64th ASIST annual meetings, v38, ASIST, 2001. p. 58–75.

Goncalves MA, Luo M, Shen R, Ali MF, Fox EA. An XML log standard and tool for digital library logging analysis. In: Agosti M, Thanos C, editors. ECDL 2002: Lecture notes on computer science, vol. 2458. Berlin: Springer; 2002. p. 129–43.

Grossmman DA, Frieder O. Information retrieval. Algorithms and heuristics. Boston: Kluwer Academic Publishers; 1998.

Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. ACM Trans Inf Syst 2004;22(1):5–53.

Hofmann T. Latent semantic models for collaborative filtering. ACM Trans Inf Syst 2004;22(1):89–115.

Hu N, Dannenberg RB. A comparison of melodic database retrieval techniques using sung queries. In: Proceedings of the joint conference on digital libraries, Portland, OR; 2002. p. 301–7.

Huang Z, Chen H, Zeng D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Trans Inf Syst 2004;22(1):116–42.

Ioannou S, Moschovitis G, Ntalianis K, Karpouzis K, Kollias S. Effective access to large audiovisual assets based on user preferences. In: Proceedings of the 2000 ACM workshops on multimedia. New York: ACM Press; 2000. p. 227–32.

Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR, Riedl J. Grouplens: applying collaborative filtering to Usenet news. Commun ACM 1997;40(3):77–87.

McFall M, Schneider K. The USA patriot act and what you can do. California Library Association Newsletter; 2003 URL ⟨http://www.cla-net.org/resources/articles/us_patriot_act.php⟩

Mobasher B, Dai H, Luo T, Nakagawa M. Effective personalization based on association rule discovery from web usage data. In: Proceeding of the third international workshop on web information and data management, New York: ACM Press; 2001. p. 9–15.

Nelson ML, Bollen J, Calhoun JR, Mackey CE. User evaluation of the NASA technical report server recommendation service. In: Sixth ACM international workshop on Web Information and Data Management (WIDM 2004), Washington, DC; 2004. p. 144–52.

Porter M. An algorithm for suffic stripping. Automated Library Inf Syst 1980;14(3):130–7.

Rocha LM, Bollen J. Biologically motivated distributed designs for adaptive knowledge management. In: Cohen I, Segel L, editors. Design principles for the immune system and other distributed autonomous systems. Oxford: Oxford University Press; 2000. p. 305–34.

Salton G. Automatic text processing. Reading, MA: Addison-Wesley; 1989.

Salton G. Term-weighting approaches in automatic text retrieval. Inf Process Manage 1998;24(5): 513–23.

Sarwar BM, Karypis G, Konstan JA, Riedl J. Analysis of recommendation algorithms for e-commerce. In: ACM conference on electronic commerce, ACM, Minneapolis, MN; 2000. p. 158–67.

Sarwar B, Karypis G, Konstan J, Reidl J. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web (WWW10). New York: ACM Press; 2001. p. 285–95.

Schafer JB, Konstan J, Riedl J. Recommender systems in e-commerce. In: Proceedings of the first ACM conference on electronic commerce, Denver, CO, 1999. p. 158–66.

Shardanand U, Maes P. Social information filtering: algorithms for automating "word of mouth". In: ACM conference proceedings on human factors in computing systems, Denver, CO; 1995. p. 210–7.

Shyu M-L, Chen S-C, Chen M, Zhang C, Sarinnapakorn K. Image database retrieval utilizing affinity relationships. In: Proceedings of the first ACM international workshop on multimedia databases. New York: ACM Press; 2003. p. 78–85.

Slaughter L, Marchionini G, Geisler G. Open Video: a framework for a test collection. J Network Comput Appl 2000;23(3):219–45.

Smeaton A, Callan J. Joint DELOS-NSF workshop on personalisation and recommender systems in digital libraries. SIGIR Forum 2001;35(1):7–11.

Sun J, Wang Z, Yu H, Nishino F, Katsuyama Y, Naoi S. Effective text extraction and recognition for www images. In: Proceedings of the 2003 ACM symposium on document engineering. New York: ACM Press; 2003. p. 115–7.

Torres R, McNee SM, Abel M, Konstan JA, Riedl J. Enhancing digital libraries with techlens+. In: JCDL '04: Proceedings of the fourth ACM/IEEE-CS joint conference on digital libraries. New York, NY, USA: ACM Press; 2004. p. 228–36.

Twidale MB, Nichols DM, Paice CD. Browsing is a collaborative process. Inf Process Manage 1997;33(6):761–83.

Van de Sompel H, Lagoze C, Nelson ML, Warner S. The open archives initiative protocol for metadata harvesting. 2002. URL ⟨http://www.openarchives.org/OAI/openarchivesprotocol.html⟩

Van de Sompel H, Young JA, Hickey TB. Using the OAI-PMH ... differently. D-Lib Magazine 2003; 9(7–8).

Wildemuth BM, Marchionini G, Yang M, Geisler G, Wilkens T, Hughes A, et al. How fast is too fast?: evaluating fast forward surrogates for digital video. In: JCDL '03: Proceedings of the third ACM/IEEE-CS joint conference on digital libraries. Washington, DC, USA: IEEE Computer Society; 2003. p. 221–30.

Woodruff A, Gossweiler R, Pitkow J, Chi EH, Card SK. Enhancing a digital book with a reading recommender. In: Proceedings of the CHI 2000 conference on human factors in computing systems, The Hague, Netherlands, 2000. p. 153–60.