

Distributed, Real-Time Computation of Community Preferences

Thomas Lutkenhouse, Michael L. Nelson, Johan Bollen

Old Dominion University
Computer Science Department
Norfolk, VA 23529 USA

{lutken,mln,jbollen}@cs.odu.edu

ABSTRACT

We describe the integration of smart digital objects with Hebbian learning to create a distributed, real-time, scalable approach to adapting to a community's preferences. We designed an experiment using popular music as the subject matter. Each digital object corresponded to a music album and contained links to other music albums. By dynamically generating links among digital objects according to user traversal patterns, then hierarchically organizing these links according to shared metadata values, we created a network of digital objects that self-organized in real-time according to the preferences of the user community. Furthermore, the similarity between user preferences and generated link structure was more pronounced between collections of objects aggregated by shared metadata values.

Categories and Subject Descriptors

H.5.4 [Hypertext/Hypermedia]: Architectures.

General Terms

Algorithms, Measurement, Performance, Design, Experimentation, Human Factors.

Keywords

Buckets, Smart Objects, Traversal Patterns, Link Computation

1. INTRODUCTION

We describe a real-time, scalable methodology to compute community preferences that can operate across multiple web servers. We have previously used Hebbian learning to capture community preferences in digital libraries, but always through batch-based log analysis of a single web server. By combining Hebbian learning with smart digital objects, we can compute the preferences in real-time. We describe an experiment that demonstrates this technique using metadata about popular music.

1.1 Buckets

The Smart Objects, Dumb Archives (SODA) model allows for many of the services previously provided by repositories to be

pushed down to the level of the digital object itself. In the SODA model, complex digital objects called "buckets" provide an interface to their contents, enforce terms and conditions, and generally assume all functionality required by the user to access the heterogeneous data stored within the object [22][23].

Rather than a mere directory used to store various file types, buckets provide their own structure for containing data and methods. A bucket contains N elements. Each element may be a discrete resource of a particular file type, or may be a set of other elements. An element may also be a URL pointer to a network object, such as another bucket. Buckets were developed in the context of managing many related data objects for NASA digital libraries. They were also designed to not depend on any particular repository implementation for their operation. Instead, they carry their required source code inside themselves, trading redundancy for increased survivability.

The self-contained and flexible nature of buckets make them amenable for use in a variety of user applications. The nested element structure imposes no fixed storage limits, and the access methods can be customized to accommodate any idiosyncratic data types. They are easily accessible in a networked environment because their methods can be invoked via HTTP requests. The following is one of the buckets in this experiment:

<http://www.cs.odu.edu/~lutken/bucket/215/>

This URL does not provide any explicit method to invoke, so the bucket defaults to its display method, producing the output shown in Figure 1. Thus, a more explicit method invocation with the same result would be:

<http://www.cs.odu.edu/~lutken/bucket/215/?method=display>

The display method presents the bucket's metadata, as shown in Figure 1, and also the bucket contents. In this experiment, the contents we are interested in managing are a set of peer buckets represented by URLs. At the bottom of Figure 1, the list of recommended LPs is shown (slightly truncated for size; 4 additional artist/LPs are present in the original web page).

1.2 Hebb's Laws of Learning

The method by which links between buckets are dynamically created and adjusted in weight is based upon Hebb's Law of Learning. Hebb's law postulates that the connection between two neurons in the brain becomes stronger when the neurons repeatedly fire in close temporal proximity [2]. In this way, the very structure of the brain is affected by previous usage patterns. This is naturally a simplified lens with which to examine the complicated system of neurological functions,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HT'05, September 6–9, 2005, Salzburg, Austria.

Copyright 2005 ACM 1-59593-168-6/05/0009...\$5.00.

but it provides a useful model that has found applications in machine learning. Hebbian learning has been successfully used in adaptive hypertext networks capable of learning usage patterns and rerouting hyperlinks accordingly [14]. As a close analogy to such a system, we use a set of rules derived from Hebbian learning to create dynamic links between smart digital objects.

The work presented here builds on our previous work in developing recommendations in scholarly digital libraries. It is important to note that in these projects, personalization is not the goal. Rather, we attempt to extract the collective preferences of a particular focused community centered around the digital library in question. In [3][6] we used Hebbian learning and log analysis techniques to make recommendations of related journals based on the reading patterns of Los Alamos National Laboratory researchers. In [20], we used similar techniques to implement and evaluate a recommendation service for the NASA Technical Report Server (NTRS). Since NTRS is open to the public and NASA cannot track the actions of individuals without a significant bureaucratic burden, the collective community preference model maps well to this application. In [4], we implemented and evaluated a recommendation server for the Open Video project. Since videos in the Open Video Project tend to have very limited textual metadata, this project demonstrated the usefulness of mining the browsing patterns of users to generate recommendations for related videos.

However, all of these previous implementations have been based on batch analysis of server logs and periodic adjustment of links, typically on a monthly basis. None of the systems could immediately show the impact of user's actions.

1.3 Related Work

There has been a significant amount of work on generating recommendations for web pages. Some studies have focused on using collaborative filtering techniques to adjust online radio playlists [9][15]. However, these techniques require the maintenance of specific user profiles, something that is not required with our technique. A number of studies have focused on improving existing recommendation systems, such as [12][17] using singular value decomposition to improve recommendations. Exploiting existing social networks to increase the quality of recommendations is described in [19]. In [25], automated and human generated recommendations were compared and although human generated recommendations were preferred, the automated recommendations were often provided interesting and novel. A comprehensive framework for evaluating recommendation systems is provided in [16].

Adaptive hypertext networks have been explored before, but in a different implementation styles. The Distributed Dynamic Link Service [8][32] implements adaptive links in a peer-to-peer network, but it requires the use of separate link servers and link bases. Passive and directed client-side link adaptation is described in [28]. User directed linking behavior has also been widely studied in a variety of contexts, such as [30][24][31]. However, none of these systems have been implemented with smart-objects similar to buckets.

Buckets are just one of many complex digital object formats that derive from the original Kahn-Wilensky Framework [18]. However, buckets are unique in that they combine the metadata, data and methods all in one storage object, and not in a repository. A full survey of complex digital objects used in digital library applications can be found in [21].

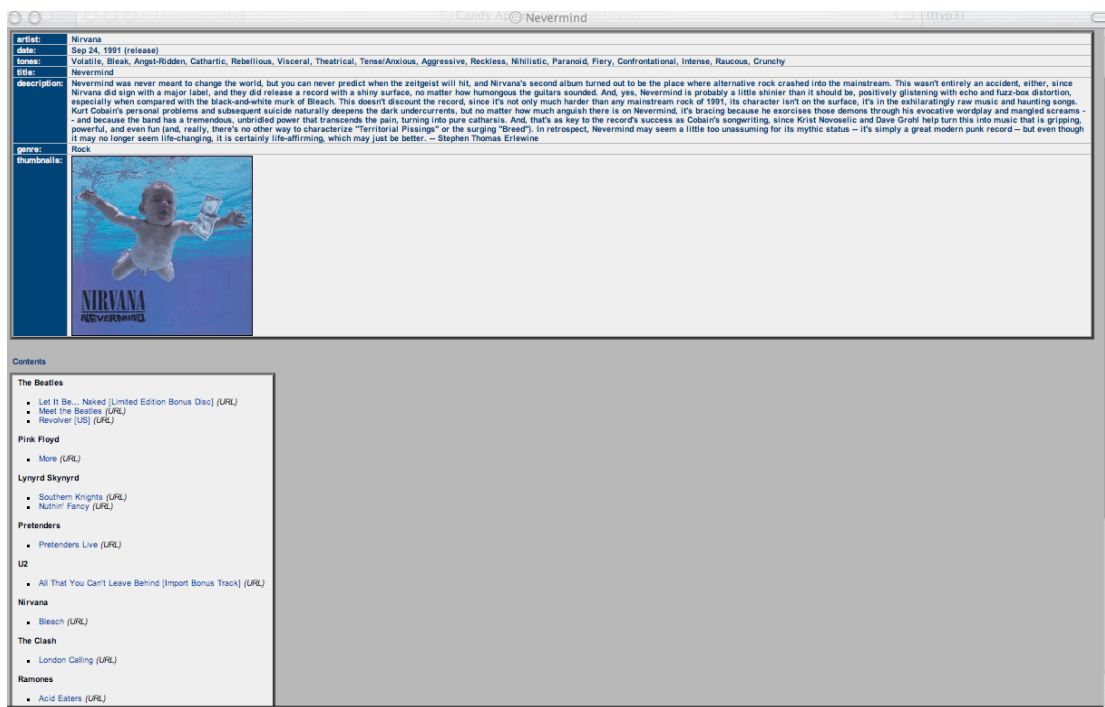


Figure 1. A bucket for the LP "Nevermind" with related LP links at the bottom.

2. IMPLEMENTING DYNAMIC LINKING

Drawing from the model of near-synchronous neuron firing strengthening the connection between the firing pair, we assume that buckets accessed in quick succession are likely related, and the link between them should therefore be strengthened. Implementing this, each bucket maintains its links to other buckets as part of its XML encoded metadata (Figure 2). In order to monitor user traversal patterns and manage the links accordingly, these URLs do not simply link to the target bucket's display method. Instead, each link points back to the current bucket, with a redirect to the target. For example, the request:

```
http://www.cs.odu.edu/~lutken/bucket/215/
?method=display&redirect=http://www.cs.odu.
edu/~lutken/bucket/29/?method=display
```

would cause the current bucket (215) to redirect to target bucket 29. By directing all requests through the current bucket, each bucket is able to detect patterns in the browsing habits of its users and dynamically update its links.

```
--<element wt="0.5"
  id="~http://www.cs.odu.edu/~lutken/bucket/29/">
  <metadata>
    <descriptive>
      <title>Let it Be</title>
    </descriptive>
    <administrative />
  </metadata>
</element>
```

Figure 2. XML fragment of a weighted link.

2.1 Updating Links

Because all requests for links are directed through the current bucket, the implicit preferences revealed when links are selected can be used to adjust link weights in a meaningful manner. Three simple rules can be implemented that reflect the principle of Hebb's Law: frequency, symmetry and transitivity. In the interest of readability, the examples in this section use "1", "2" and "3" instead of full bucket URLs.

When a user accessing bucket 1 clicks on a link to bucket 2, it can be inferred that some users who are interested in bucket 1 are also interested in bucket 2, and the link between them should therefore be strengthened. Thus, a frequency constant is added to the weight of this link every time a user makes this decision. In choosing bucket 2, the following URL will be selected:

```
http://1?method=display&referrer=1&redirect
=http://2?method=display%26referrer=http://1
```

At that time, bucket 1 will add the frequency constant to its internal representation of the link, as in Figure 3. Notice the URL also provides bucket 2 with a referrer argument. Using this argument, bucket 2 is also able to recognize the user's choice to access both of these objects in succession, and can reciprocate the link strengthening by adding a symmetry constant, as shown in Figure 4. If no link previously existed from bucket 2 to bucket 1, such a link is created at this time.

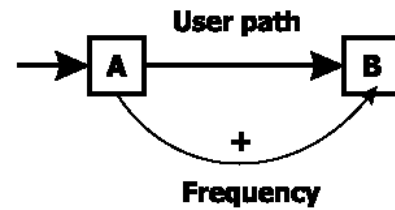


Figure 3. Addition of the frequency constant.

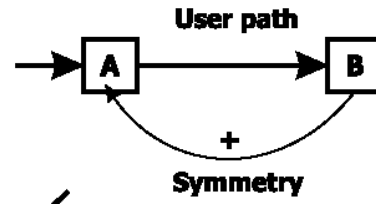


Figure 4. Addition of the symmetry constant.

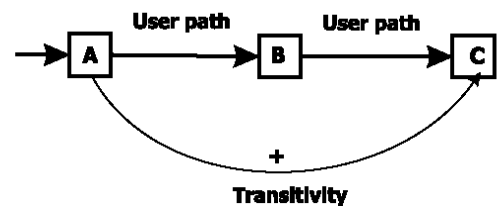


Figure 5. Addition of the transitivity constant

Finally, a mechanism is required by which buckets are able to adjust their links to each other even if they are not accessed in immediate succession. This is accomplished by recognizing that a transitive relationship may be defined by a user's path through a set of buckets. For instance, a path from bucket 1 to bucket 3 via bucket 2 demonstrates not only a relationship between buckets 1 and 2, or buckets 2 and 3, but also a less direct relationship between buckets 1 and 3.

Following such a path, the URL generated in bucket 2 to allow the traversal to continue to bucket 3 would have the following structure:

```
http://2?method=display&referrer=2&redirect
=http://1?method=display%26redirect=http:
//3?method=display%26referrer=http://2
```

Because bucket 2 is shown to be the referrer, and the ultimate redirect is for bucket 3, bucket 2 increments its own link weight by the frequency constant as in the previous example. However, instead of an immediate redirect to bucket 3, we have an intermediate redirect to bucket 1. The display method of bucket 1 is invoked without any referrer argument. Because this argument is missing, bucket 1 is able to recognize this as a transitive operation, and therefore increments its link weight to bucket 3 by the transitivity constant (Figure 5). If a direct link from bucket 1 to bucket 3 did not exist prior to this operation, it is now created (transitivity is the only way in

which new links are introduced to the system). Bucket 3 is recognized as the object whose link should be increased because it is the target of the redirect. Note that, although a bucket 1 method is invoked by a URL generated at bucket 2, there are no frequency or symmetry adjustments made to the links between buckets 1 and 2 in this operation. The redirect from bucket 1 to bucket 3 shows bucket 2 as the referrer, so bucket 3 is able to take the appropriate step of incrementing its link to bucket 2 by the symmetry constant. As in the previous operations, if the symmetric link from bucket 3 to bucket 2 did not exist, it will now be created. This concludes the transitive operation.

2.2 Hierarchical Display of Links

The buckets use the link weights to build a hierarchical adaptive interface. The first step in this process is accessing the metadata of each digital object targeted by the bucket's links. The various metadata fields offer different opportunities to organize the links by common characteristics, but we will here assume that one metadata field has been chosen as most appropriate, and continue accordingly. Retrieving the value of this field for each target object, we use these values as the "bins" into which each link can now be grouped. The weights of the links in each bin are totaled, and the normalized weight of each bin is calculated from its weight total and the total number of links in the bin. The bins are now sorted by normalized weight, and presented to the user in decreasing order. Thus the higher each grouping appears, the more recommended its target content. The links within each bin are also presented in order of decreasing weight.

The process of hierarchically organizing links makes possible a great deal of additional functionality at the level of the digital object. For example, to spot temporal trends in research, we can group articles according to publication date. To find experts on a given topic, we can group according to author. Limited only by the available metadata, the combination of dynamically ranking objects according to user traversals and grouping those objects according to their explicit commonalities provides a powerful and flexible method for discovering meaningful connections among data objects.

3. EXPERIMENTAL TESTBED

It is infeasible to expect each digital object in a network to maintain extensive personalized logs to determine the content preference of any given user. Instead, the approach taken in this experiment is to draw from the traversal patterns of the entire user community [10] to discover the peer objects commonly co-accessed with the current object. In this way, links to the peer objects can be created and weighted so as to provide recommendations based on the combined experience of previous users [5][2]. To refine this process, we can take advantage of any metadata the digital objects have been implemented to provide. By grouping the dynamic links according to a shared value in any given metadata field, and normalizing the combined weights accordingly, we can make recommendations not only for the object level of discourse, but also for aggregate entities.

To demonstrate that the methodology outlined above could be used to evolve a network of smart objects to correspond to the preferences of a community, we created an experiment that used 800 buckets, each one representing an album released by one of the top 50 rock bands of all time as chosen by Spin

Magazine [26]. Each bucket's display method presented a brief description of the album, along with an album cover thumbnail and metadata fields of artist, release date, tones, title and genre (Figure 1). All of the album information was harvested from allmusic, a popular music website [1].

The initial links between buckets were generated randomly, with each bucket containing 5 links to other buckets, and having 5 other buckets containing links to it. All initial links were given a weight of 0.5. Link weights were increased by 1.0 for frequency, 0.5 for symmetry, and 0.3 for transitivity. These weights were empirically determined in previous research [2]. These three operations were the only means by which link weights were altered during the course of this experiment.

The metadata field used to aggregate individual buckets into the hierarchical display was the 'artist' field. Thus, the adaptive display attempted to recommend the most appropriate artist to the current bucket, and individual albums were sub-ranked therein.

An open invitation was sent in August 2004 for users to traverse the system (although this experiment was based only on user traversals, the bucket displays can be indexed by web crawlers). Users self-identified and were then directed to a random initial bucket to begin their traversal. When the experiment concluded in October 2004, 160 users had navigated the system, visiting an average of 7.5 buckets each, for a total of 1200 bucket-to-bucket traversals. At any point in their traversal, they were able to exit browsing session and rate their assessment of the link utility. Table 1 summarizes the profiles of the volunteers that formed the community for this experiment.

Table 1. Profile of the 160 Volunteers

Nationality	1 Brazil, 1 Portugal, 4 Canada, 10 UK, 20 Belgium, 124 US
Sex	124 Male, 36 Female
Age	High 72, Low 7, Average 37
Domain Knowledge Self-Assessment (1=low, 7=high)	Average = 4.0
Assessment of link utility (1=low, 5=high)	Average = 2.8

4. RESULTS

The utility of this system can be judged by how well it has incorporated community preferences into the dynamically generated link structure. Specifically, we are looking for verification that the user-defined qualities of importance and relatedness can be determined by analyzing the state of the network after users have extensively navigated the network.

4.1 Importance

Several metrics can be used to determine the importance the user community has given each music album represented as a bucket in this system. Because the system is designed to increase the number and the weight of the links connecting heavily trafficked objects, the social network analysis measurements of degree centrality and weighted degree centrality are appropriate metrics. PageRank is also an especially suitable metric, as it was designed to determine the importance of hyperlinked pages on the World Wide Web [7].

The importance of each bucket in the system was calculated according to these measures and ranked accordingly. The next step in validating the network of links was to take the rankings determined by degree centrality, weighted degree centrality and PageRank and compare them to an authoritative ranking in the music industry.

The Recording Industry Association of America (RIAA) publishes a list of the best-selling albums of all time [27]. However, only 51 of the 800 albums represented in this experiment were listed among the RIAA's best-selling albums. Comparing the ranked list of albums created by our system with the RIAA list of best-selling albums, it became apparent that those albums highest ranked by degree centrality, weighted degree centrality and PageRank were all more likely to appear on the RIAA list than those in the collection at large. Although less than 6.5% of the albums in the entire collection were represented in the RIAA best-sellers, at least 25% of those albums ranked in the top 2% by any of the three given metrics appeared on the best-seller list. Figure 6 shows this trend graphed as probability of being included as a best-seller against rank by centrality and PageRank. Although not definitive validation of importance, this trend suggested a

possible relationship between sales figures and our ranking metrics, a relationship that is further explored herein.

Because the system is designed to offer link recommendations based on content aggregated by artist, it is useful to determine whether the system has represented users' choices at the aggregate level. Indeed, because each user is much more likely to come across a favorite artist in the course of their navigation than a favorite particular album, it seems likely that aggregate preferences would be more apparent than those at the lowest level of discourse. Measures of aggregate importance were calculated by taking a mean degree centrality per bucket of a given artist. The same was done for weighted degree centrality and PageRank.

Using the aggregate metrics, the artists were also ranked and compared against the RIAA list of best selling artists. Once again, only a small number were represented among the RIAA statistics, 14 of the 50 artists included in our system. However, among the highest ranked artists by any one of our metrics, 4 out of the top 5 were found to be included in the published top seller list. Again, an interesting trend to be further explored, as shown in Figure 7.

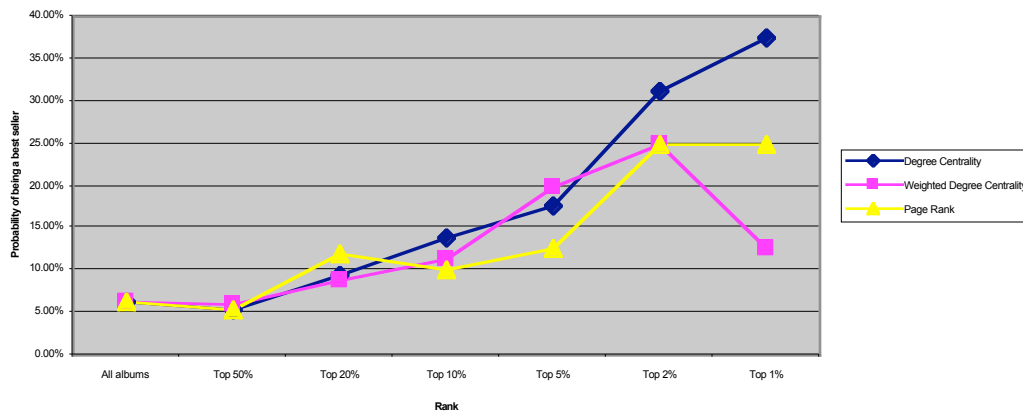


Figure 6. Probability of albums being best-sellers.

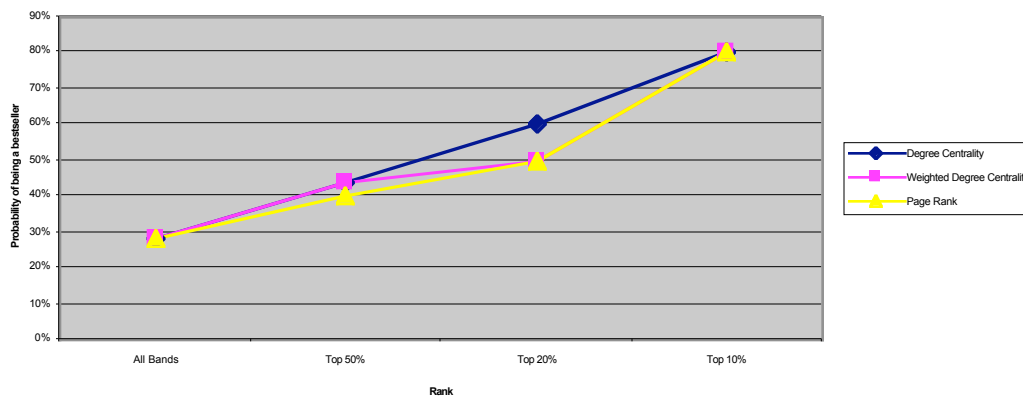


Figure 7. Probability of artists being best-sellers.

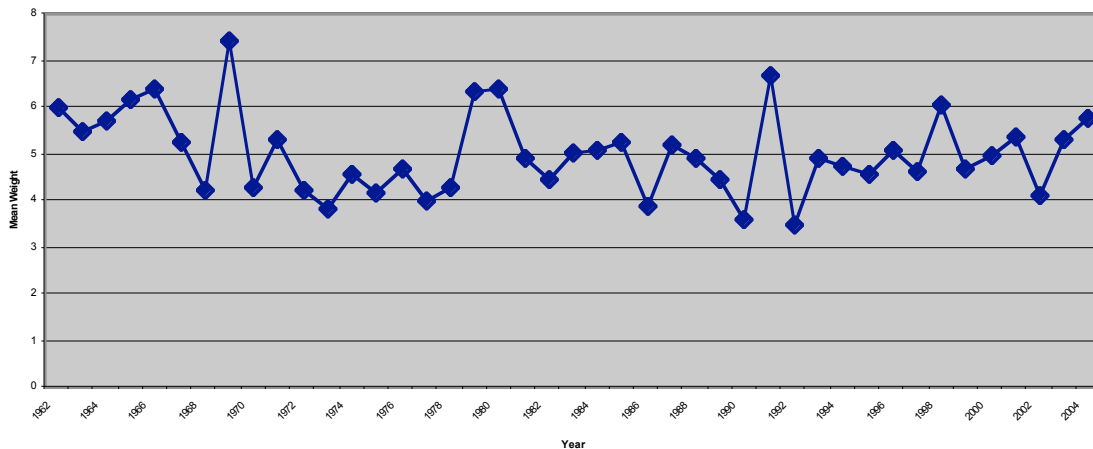


Figure 8. Mean weighted degree centrality per bucket for each year.

Using the aggregate ranking data, a more formal comparison of the system rankings to published rankings was conducted by calculating a Spearman Rank Correlation Coefficient against various published lists. This generates a coefficient between -1 and 1 , where the greater the difference from zero, the greater probability of a strong connection between the two data sets.

Using this technique, a Spearman Rank Correlation Coefficient was calculated between the artists' ranks based upon their aggregate measures of centrality and artist rankings published by VH1 [29], DigitalDreamDoor.com [11], and even the original Spin list from which the artists in this experiment were taken. Surprisingly, no significant correlation was found to exist between any of these qualitative artist rankings and the artist rankings determined by the system. The highest correlation between any of these published lists and the system rankings was 0.29 , when compared with the VH1 list of top artists. However, with 50 data pairs being compared, this coefficient has a greater than 5% likelihood of occurring by pure chance, and is therefore useless. The rankings generated by the system simply could not be shown to correlate to any of the given expert rankings.

Moving away from expert opinions and back to ranks derived from sales figures, we recorded the Amazon.com sales rank for 737 of our 800 albums. Amazon could offer no sales data on the remaining 63 albums, and they were not included in the following calculations. At the lowest level of discourse, a correlation coefficient was calculated between the Amazon sales ranks and the system-generated album ranks. These coefficients ranged from -0.11 for degree centrality to -0.18 for PageRank. Because of the large number of data pairs, it could not be conclusively determined whether the relationship between the data sets was significant.

However, comparing the aggregate data yielded more convincing results. Using the aggregate measures of degree centrality, weighted degree centrality and PageRank, the system-determined artist rankings were compared against each artist's mean Amazon sales rank. Spearman Rank Correlation Coefficients of -0.49 , -0.46 and -0.45 were

calculated, respectively. For the 50 data pairs being tested in each case, these values produce a significance of greater than 0.99 ; a less than 1% likelihood of chance occurrence. The implicit user preferences revealed by the purchasing community of Amazon.com seem to be reflected by the implicit user preferences revealed by the browsing choices of our community of users.

Measurements were also taken to determine if the community of users tended to favor any particular time period. The mean weighted degree centrality per bucket for each album release year included in this experiment was calculated. These ranged from 1962 to 2004, as shown in Figure 8. User preferences do not seem concentrated in any particular period. However, given the longevity of many of the albums (e.g., Pink Floyd's "Dark Side of the Moon"), release date and the duration of popular interest are not necessarily correlated.

4.2 Relatedness

The other half of validation is determining if related contents grew stronger links to each other in the course of users surfing the system. A principal component analysis was performed on an adjacency matrix representing all the weighted links between the artists included in this experiment. Cosine differences were calculated to build a covariance matrix, from which eigenvalues and eigenvectors were computed. Using the most significant eigenvalues, the corresponding eigenvectors were selected as the basis of a two-dimensional space in which each artist could be placed. Multiplying the covariance matrix by the eigenvectors, the two-dimensional coordinates for each artist were calculated. The results are shown in Figure 9.

The primary component analysis can only determine the mathematically most significant dimensions in which to compare the values in the adjacency matrix, it naturally cannot determine the qualitative features of the elements that caused the users to create the weighted links as they did. The accuracy of describing the vertical axis in terms of niche audience vs. general audience, and the horizontal axis in terms of originality of sound is a matter of opinion. Because no explicit or inarguable patterns were revealed, this primary

component analysis did not conclusively demonstrate any tendency toward greater relatedness of the system content.

In another analysis used to determine relatedness, a hierarchical clustering dendrogram was constructed. By taking the cosine differences between each pair of rows in an adjacency matrix representing the artists in the experiment, and grouping the artists in steps representing increasing

cosine difference, the dendrogram in Figure 10 was created. As with the PCA, no explicit patterns were revealed in this analysis. Although some individual clusters seemed to be grouped on date, such as the prominent 1970s cluster of Led Zeppelin, Bob Marley, Kiss, Lynyrd Skynyrd and The Who, there were no consistent patterns to explain the groupings of the dendrogram as a whole.

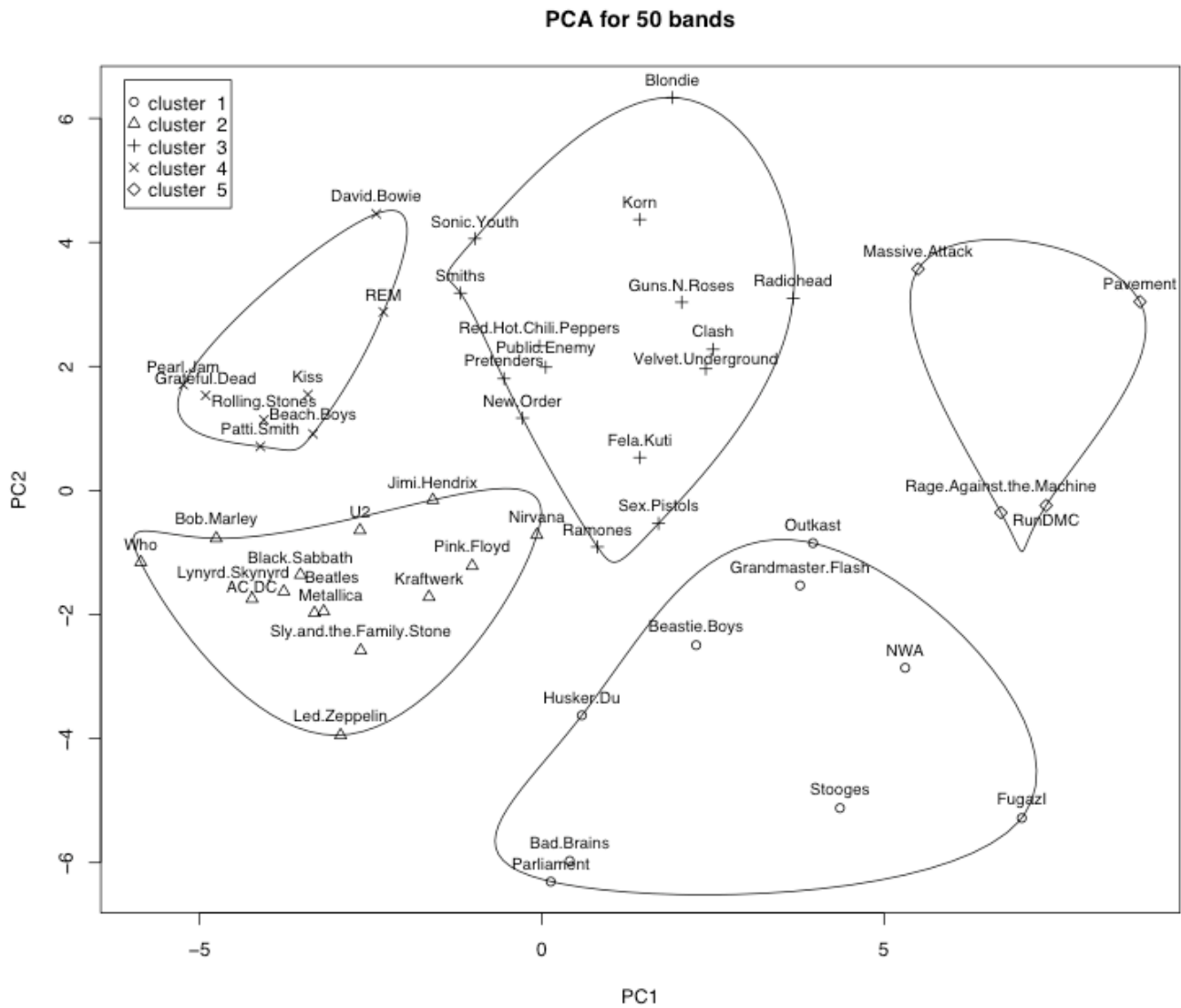


Figure 9. Two-dimensional plot of PCA.

5. FUTURE WORK

This experiment uses only one field of digital object metadata in its attempt to create a useful, dynamically generated set of recommended links. It would be interesting

to give users the option of grouping the recommendations according to their field of choice, such as date or genre.

The use of an initially random linked structure places a large burden on the early users of the system to wade through arbitrary content in search of their objects of interest. This also leads to the generation of spurious links, as the users

blindly click from one bucket to the next in search of interesting content. This might be addressed by initially linking the content according to an explicit indexing scheme (e.g., musical genre, release date), and then testing if dynamic hierarchical link generation is able to add value to an already coherent network.

Another approach might attempt to collect explicit user assessments in tandem with the implicit preferences shown by browsing habits. For example, the user might be presented with a set of links for each related digital object, each one of which serves the dual purposes of grading the current bucket while linking to the next bucket.

As in any user evaluation study, more participation is always desirable. Although 160 users and 1200 link traversals was satisfactory, more users and more link traversals would have

resulted in a more stable system over time. If the testbed were to be kept in place for a longer running experiment, we would have needed to add a link attrition policy, such as limiting the display to the first 10 top-level links or slowly decrementing the value of all links.

Finally, a larger number of bands, including bands representing a larger variety of musical genres might lend itself to more concrete patterns emerging after sufficient user traversal. The original list of 50 bands shows some genre diversity, but is limited to the bands of interest to Spin Magazine readers. A future version of this work would combine the Spin Magazine artist list with similar lists from Down Beat (Jazz), Vibe (Urban), Music City News (Country), etc.

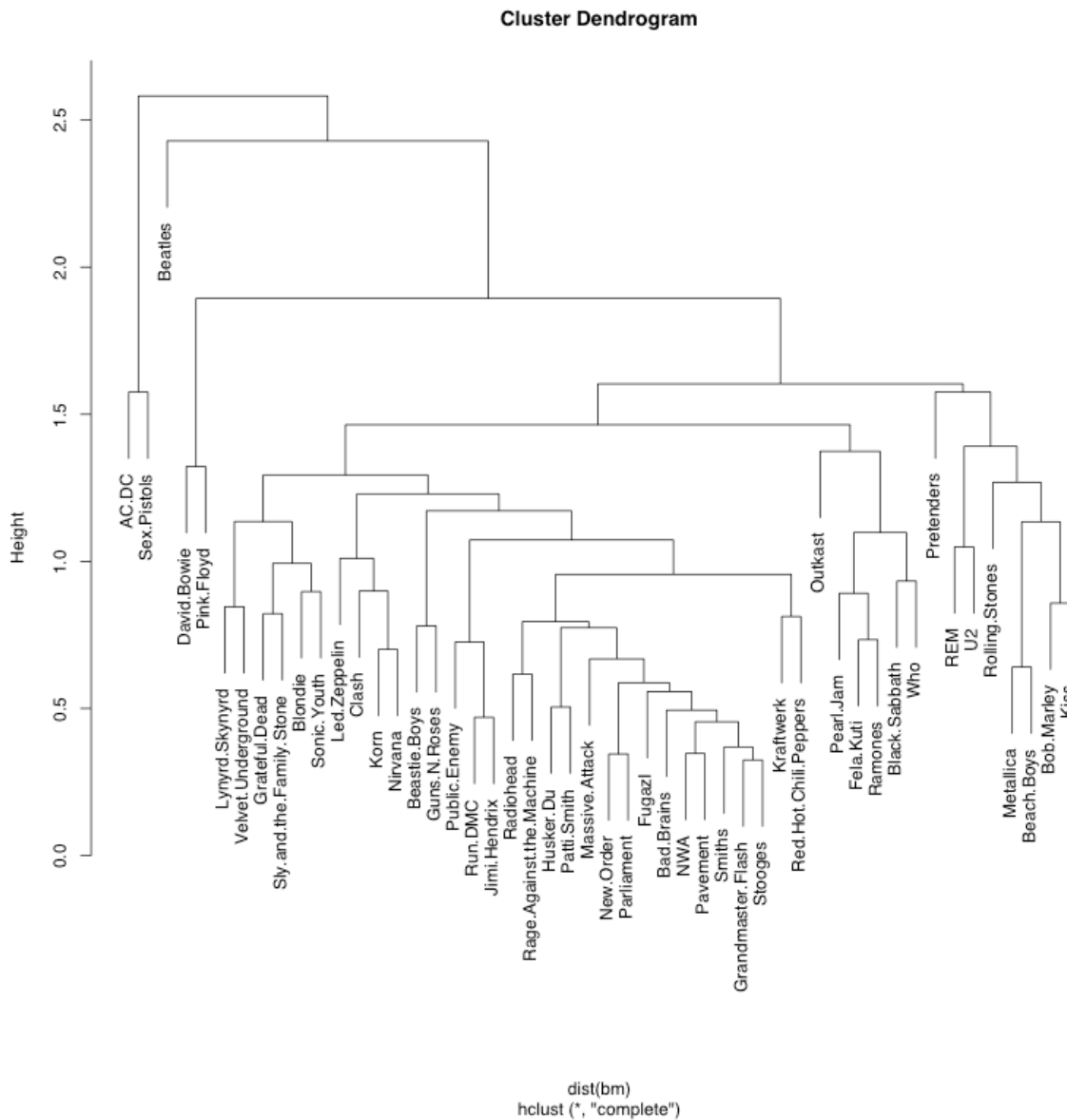


Figure 10. Cluster dendrogram of the 50 original artists.

6. CONCLUSIONS

Previous implementations of Hebbian learning recommendation services have been based on log analysis and the links were only updated periodically (generally monthly). Link computation and update was an expensive, slow process. We have described the implementation of a based recommendation service in a network of smart objects. The smart objects allow for the same computation of links, but in a scalable, distributed and real-time manner.

The network of smart objects built for this experiment demonstrated an ability to monitor usage patterns, and adjust its inter-object links according to that usage. User traversals completely altered the topology of the network from an unbiased initial state with equally weighted links, to a system of objects greatly differentiated by their dynamically generated positions of centrality.

Comparing the system-generated importance of the objects in the network, as measured by degree centrality, weighted degree centrality and PageRank, against any published list of expert opinion proved fruitless. The implicit opinions of the experiment's user community simply did not match those of any expert we found. The preferences expressed by sales rank, however, proved a much better match. The relationship between user preferences and sales rank was shown to be strong, especially when measured in the aggregate.

Increased relatedness among the objects in the network could not be demonstrated. There are several possible causes of this. The quantity of network traffic was likely insufficient to fully develop links demonstrative of users' perceived relations between objects. The initial random structure of the network also forced early users to generate meaningless links as they clicked in search of their content of interest. These links likely obscured actual content relations. Lastly, the content of the objects in this experiment may have been "pre-selected" for relatedness relative to the readership of Spin Magazine, making coarse patterns (e.g., jazz, rock, country, etc.) hard to uncover.

7. ACKNOWLEDGMENTS

We thank the 160 volunteers who participated in this study. We also thank Aravind Elango, who performed an earlier version of this experiment.

8. REFERENCES

- [1] All Music Guide's Allmusic.com
<http://www.allmusic.com>
- [2] J. Bollen, F. Heylighen. A system to restructure hypertext networks into valid user models. *The New Review of Hypermedia and Multimedia*, 4, 1998, 189-213.
- [3] J. Bollen and R. Luce, Evaluation of digital library impact and user communities by analysis of usage patterns, *D-Lib Magazine*, 8, 6, 2002.
- [4] J. Bollen, M. L. Nelson, R. Araujo, G. Geisler. Rapid prototyping of a multimedia recommender system. In *Proceedings of JCDL 2005*, (June 7-10, 2005, Denver, CO), p. 369.
- [5] J. Bollen and M.L. Nelson. Adaptive Networks of Smart Objects. In *Proceedings of Workshop on Distributed Computing Architectures for Digital Libraries (ICPP02)*, (August 18-21, 2002, Vancouver, Canada), pp. 487-496.
- [6] J. Bollen, S. Vemulapalli, W. Xu. Digital library evaluation by analysis of user retrieval patterns. In *Proceedings of ECDL 2002*, (September 2002, Rome, Italy), pp. 432-448.
- [7] S. Brin, L. Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30, 1-7, 1998, 107-117.
- [8] L. A. Carr, D. C. De Roure, W. Hall, and G. J. Hill. The Distributed Link Service: A tool for Publishers, Authors and Readers. *World Wide Web Journal*, 1, 1, 1995, 647-656.
- [9] D. L. Chao, J. Balthrop, and S. Forrest. *Adaptive Radio: Achieving consensus using negative preferences*. Technical Report TR-CS-2004-08, The University of New Mexico Department of Computer Science, Albuquerque, NM, 2004.
- [10] D. Crow, B. Smith. The role of built-in knowledge in adaptive interface systems. In *Proceedings of the First International Conference on Intelligent User Interfaces*, (January 4-7, 1993, Orlando, FL), pp. 97-104.
- [11] Digital Dream Door, <http://www.digitaldreamdoor.com>
- [12] M. Efron, G. Geisler, Using Dimensionality Reduction to Improve Similarity Judgments for Recommendation, In *Proceedings of the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries*, (Dublin City University, Ireland, June 18-20 2001).
- [13] Aravind Elango, Johan Bollen and Michael L. Nelson. Dynamic Linking of Smart Digital Objects Based on User Navigation Patterns. arXiv Technical Report cs.DL/0401029, January 2004.
<http://www.arxiv.org/abs/cs.DL/0401029>
- [14] D. O. Hebb, *The Organization of Behavior*. John Wiley, New York, 1949.
- [15] D. Hauver, J. French. Flycasting: Using Collaborative Filtering to Generate a Playlist for Online Radio. In *Proceedings of First International Conference on Web Delivering of Music*, (November 23 - 24, 2001 Florence, Italy), 123-130.
- [16] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. Riedl, Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22, 1, 2004, 5-53.
- [17] T. Hofmann, Latent semantic models for collaborative filtering, *ACM Transactions on Information Systems*, 22, 1, 2004, 89-115.
- [18] R. Kahn, R. Wilensky, A Framework for Distributed Digital Object Services, cnri.dlib/tn95-01, 1995.
- [19] H. A. Kautz, B. Selman, M. A. Shah, Referral Web: Combining Social Networks and Collaborative Filtering, *Communications of the ACM* 40, 3, 1997, 63-65.
- [20] M. L. Nelson, J. Bollen, J. R. Calhoun, C. E. Mackey. User Evaluation of the NASA Technical Report Server Recommendation Service. In *Proceedings of WIDM 2004*, (November 12-13, 2004, Washington DC), 144-151.
- [21] M. L. Nelson, B. Argue, M. Efron, Sheila Denn, M. C. Pattuelli. A Survey of Complex Object Technologies for

- Digital Libraries, NASA/TM-2001-211426, December 2001.
- [22] M. L. Nelson, K. Maly. Buckets: Smart Objects for Digital Libraries, *Communications of the ACM*, 44, 5, 2001, 60-62.
- [23] M. L. Nelson, K. Maly, M. Zubair, S. N. T. Shen. SODA: Smart Objects, Dumb Archives. In *Proceedings of ECDL 1999*, (September 22-24, 1999, Paris France), pp. 453-464.
- [24] M. N. Price, G. Golovchinsky, B. N. Schilit, Linking by inking: trailblazing in a paper-like hypertext, In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*, (June 20-24, 1998, Pittsburgh, PA), pp. 30-39.
- [25] R. R. Sinha, K. Swearingen, Comparing Recommendations Made by Online Systems and Friends, In *Proceedings of the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries*, Dublin City University, (Ireland, June 18-20 2001).
- [26] Top 50 Music Bands of All Time. Spin, February 2002. <http://www.rocklist.net/spin100.html>
- [27] Top 100 Albums / Artists
<http://www.riaa.com/gp/bestsellers/>
- [28] T. Tsandilas and m. c. schraefel, User-controlled link adaptation, In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, (August 26-30, 2003, Nottingham, UK), pp. 152-160.
- [29] VH1: 100 Greatest Artists of Rock & Roll, <http://www.rockonthenet.com/archive/1998/vh1artists.htm>
- [30] H. Wu, M. D. Gordon, K. DeMaagd and N. Bos, Link analysis for collaborative knowledge building, In *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, (August 26-30, 2003, Nottingham, UK), pp. 216-217.
- [31] T. Yan, M. Jacobsen, H. Garcia-Molina and U. Dayal, From User Access Patterns to Dynamic Hypertext Linking, In *Proceedings of the Fifth International World Wide Web Conference*, (May 6-10, 1996, Paris, France).
- [32] J. Zhou, W. Hall, and David De Roue, When Open Hypermedia Meets Peer-to-Peer Computing, In *Proceedings of the Fifteenth ACM Conference on Hypertext & Hypermedia*, (August 9-13, 2004, Santa Cruz, CA), pp. 266-267.