Discovering Hidden Contextual Factors for Implicit Feedback

Massimo Melucci[1] and Ryen W. White[2]

[1] University of Padova
Department of Information Engineering
Via Gradenigo 6 — 35131 – Padova, Italy
melo@dei.unipd.it
[2] Microsoft Research
One Microsoft Way
Redmond, WA 90852 USA
ryenw@microsoft.com

Abstract. This paper presents a statistical framework based on Principal Component Analysis (PCA) for discovering the contextual factors which most strongly influence user behavior during information-seeking activities. We focus particular attention on explaining how PCA can be used to assist in the discovery of contextual factors. As a demonstration of the utility of PCA, we employ it in an Implicit Relevance Feedback (IRF) algorithm that observes features of user interaction, computes the feature co-variances from a few seen documents, and calculates the eigenvectors of the co-variance matrix to be used as the basis for ranking the unseen documents. This ranking is then compared with the ideal ranking that could be computed if the ratings explicitly given by the user were known. The most effective eigenvector, in terms of impact on retrieval performance, was chosen as representative of each user's intent. Our experiments showed that each aspect of user behavior is influenced by different contextual factors, yet there exist some important features common to most factors. Our findings demonstrate both the effectiveness of the IRF algorithm and the potential value of incorporating multiple sources of interaction evidence in their development. In particular, it was shown that IRF was more effective when the eigenvectors are personalized to each user.

1 Background

Users with vague information needs or limited search experience often require ways to make their queries more precise. Relevance Feedback (RF) [1] provides an effective way of doing this by using relevance information explicitly provided by users. However, despite the promise of RF, users are reluctant to provide explicit feedback, generally because they do not understand its benefits or do not perceive it as being relevant to the attainment of their information goals [2]. As an alternative, implicit RF (IRF) [3] uses features of the interaction between the user and information (e.g., the amount of time a document is in focus in the Web browser or on the desktop, saving, printing, scrolling, click-through), where visited documents to which certain relevance criteria apply are assumed to be relevant. Such contextual features can be mined and used as the basis for relevance criteria in IRF algorithms. These algorithms can suggest query expansion terms, retrieve new search results, or dynamically reorder existing results.

The approach we describe in this paper utilizes user behavior features observed from interaction. Much of the research in this area has focused on the impact on the reliability of interaction features of task and user information [4, 5], click-through [6], session duration and number of result sets returned [7], and document display time [4]. These studies showed that the combination of several implicit features, including display time and the way the user exited from the result page, can predict search result relevance. However, they have also shown that interpreting click-throughs as absolute relevance judgments is sometimes difficult, display times differ significantly according to specific task and user, and that factors such as task, user experience, and stage in the search can affect the potential usefulness of IRF.

As well as developing a better understanding of the accuracy of IRF and the factors that can affect it, work is also ongoing in using this feedback to develop more advanced search systems. Researchers have explored issues such as how behaviors exhibited by users while reading articles from newsgroups could be used as IRF for profile acquisition and filtering [8], to develop a system capable of automatically retrieving documents and recommending URLs to the user based on what the user was typing in a non-search application [9], and to automatically re-rank sentence-based summaries for retrieved documents [10]. To perform these and similar functions IRF has generally been limited to a single behavior such as document display time, editing, or visitation [11–13]. Multiple aspects of user interaction behavior have also been employed [14], but not in the search domain.

In this paper we use multiple aspects of user interaction behavior during search to build models of user interests that can be useful in ranking documents as yet unseen by the user. In the remainder of this paper we describe the approach we adopt and its application for IRF in Section 2, the experiment performed to test its value and its findings in Section 3, and conclude in Section 4.

2 Discovering Hidden Contextual Factors

Information-seeking and retrieval activities are affected by contextual factors that cannot be modeled directly. A *contextual factor* is a variable (e.g., user behavior) that describes one of the possible ways context affects user activity. The *features* are the data observed from user activity. Suppose an observer is trying to understand user behavior when the user is seeking information by measuring various features (e.g., document display time, amount of scrolling). The observer wants to build a model of the user's behavior for modifying the system so as to associate the most relevant documents to that model. Unfortunately, he cannot figure out what is happening because the features appear clouded, sometimes redundant or missing. If a model of user behavior *exists*, then it is hidden behind clouds of noisy data. “Hidden” is related to the latent variables which could not have been observed directly because of the ignorance of the real structure of the information-seeking and retrieval activities. For example the amount of scrolling is likely related to document display time, and therefore one of the two features are probably redundant. What the observer does not know is the degree of redundancy or if some other unobservable variable is governing both features — it may be that this unobservable variable is related to both features so as to make them co-related although they are not when the unobservable is absent. Since these factors are hidden, a mechanism for extracting them is necessary. In this section we present a statistical framework based on Principal Component Analysis (PCA) that can be used to represent these factors in a way that can be leveraged by IR systems for improved retrieval effectiveness.

A statistical framework can discover hidden information from amounts of noisy data [15]. One reason the observed data may be noisy is the absence of (meta-)data about the contextual factors from which the data were observed. In other words, the factors which explain the data are hidden and noise is what makes the data not perfectly explained by the factors. This would mean that a naïve perspective has been taken when observing the data, that is, a perspective for which no noise would exist, and therefore no data about context has been observed. It may be that, if this perspective is changed, noise can be reduced if not removed and the hidden factors governing information-seeking activities can emerge. In the following, we show how the statistical framework can be used for changing perspective and discovering the hidden factors.

When the statistical framework is adopted, the data are naturally represented as vectors and matrices — vector spaces and Linear Algebra is the theoretical framework on which the Vector Space Model for IR (VSM) was proposed in the early Seventies and some recent advances on modeling IR and in particular IR in Context was investigated [16–19, 21]. Thus, when a document or Web page is visited, a feature vector can be associated to it. If k features are observed for each document, the document vectors exist in a k-dimensional vector space. Linear Algebra tells us that every vector in k-dimensional vector space can be represented as a linear combination of k independent *basis* vectors.[1]

In [16, 17], the idea that a document feature vector is the result of a linear combination of basis vectors for representing a document as the result of a combination of hidden contextual factors was presented. Therefore, the discovery of the basis vectors which have generated a document feature vector permits to have a representation of the contextual factors which explain why those features have been observed. With this in mind, suppose a document feature vector has been observed. What is the basis? That is, what are the factors?

This question is important because it points out a tacit assumption which is often overlooked. Indeed, the basis assumed is often the canonical basis. For example, $\{(1,0),(0,1)\}$ is the canonical basis of the two-dimensional vec-

[1] A set of vectors are mutually independent if no vector is a linear combination of the others.

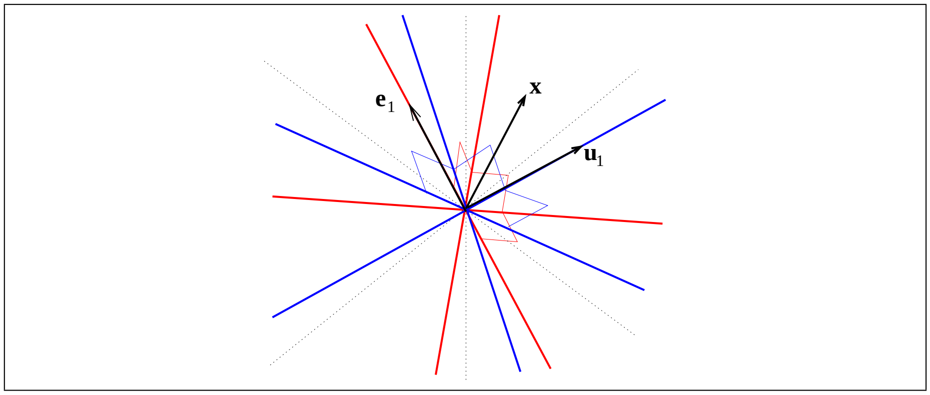

Fig. 1. A vector is generated by infinite dimensions.

tor space and every vector of this space is expressed as linear combination of the canonical vectors. However, nothing prevents us from expressing the same document feature vector as a linear combination of a different basis such as $\left\{\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right), \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)\right\}$ — the vector is the same but its coordinates are different.

In Figure 1 we show how the framework represents a document seen from two perspectives given by two bases. There are two sets of rays — one set of rays is spanned by the basis $E = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, while the other set is spanned by the basis $U = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$. Figure 1 depicts how many contextual factors are in the same space. This superposition of factors can naturally be represented by the infinite sets of coordinates which can be defined in the vector space. In the figure, E superposes U. Both E and U can "generate" the same vector $\mathbf{x}$. The myriad of bases model a document or a query from different perspectives and each perspective corresponds to a distinct set of contextual factors. Mathematically, a vector $\mathbf{x}$ is generated by the contextual factors $\{\mathbf{u}_1, \mathbf{u}_2\}$ as $\mathbf{x} = p_1^2\mathbf{u}_1 + p_2^2\mathbf{u}_2 + p_2^3\mathbf{u}_3$ where $\mathbf{u}_i \perp \mathbf{u}_j$, $i \neq j$, $p_1^2 + p_2^2 + p_3^2 = 1$ and $p_i^2 \geq 0$. At the same time, $\mathbf{x} = q_1^2\mathbf{e}_1 + q_2^2\mathbf{e}_2 + q_3^2\mathbf{e}_3$ where $\mathbf{e}_i \perp \mathbf{e}_j$, $i \neq j$, $q_1^2 + q_2^2 + q_3^2 = 1$ and $q_i^2 \geq 0$. An explanation of these expressions is given in [19].

This can be expressed in Linear Algebra using matrices. Let

$$\mathbf{E} = \begin{bmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_k \end{bmatrix} = \mathbf{I}$$

be the matrix of the canonical basis, which is the starting point of the analysis, that is, the document feature vectors have been observed in this basis. The question is: *Is there another basis which expresses the same vectors and at the same time describe the hidden factors?* The answer is provided by PCA which yields a matrix $\mathbf{C}$ that transforms the feature vectors expressed in the canonical basis into vectors expressed in the new basis.

Mathematically, the change of basis is as follows. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ be the $n \times k$ document feature matrix where $\mathbf{x}_i$ is the i-th (row) document feature vector — the column vectors of $\mathbf{X}$ are supposed to have zero means. Let $\mathbf{V} = \mathbf{X}^\top \cdot \mathbf{X}$ be the feature co-variance matrix.[2] PCA yields $\mathbf{C} = \mathbf{U}$ where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$ is the matrix of the eigenvectors of $\mathbf{V}$. The transformation takes place as $\mathbf{Y} = \mathbf{X} \cdot \mathbf{C}$ so as $\mathbf{y}_i = \mathbf{x}_i \cdot \mathbf{C} = [\mathbf{x}_i \cdot \mathbf{u}_1, \ldots \mathbf{x}_i \cdot \mathbf{u}_k]$. One can recognize that the j-th coefficient of $\mathbf{y}_i$ is the size of the projection of $\mathbf{x}_i$ on to the j-th eigenvector. Because the eigenvectors are mutually orthonormal,

$$\mathbf{x}_i^\top = (\mathbf{x}_i \cdot \mathbf{u}_1)\mathbf{u}_1 + \ldots (\mathbf{x}_i \cdot \mathbf{u}_k)\mathbf{u}_k$$

Therefore, the eigenvectors are the new basis in which the document feature vectors are expressed and are the representation of the contextual factors underlying the generation of the document feature vectors.

Why is PCA so special? In statistics, PCA is used to find the vector of features which best explains the variance of the data. To obtain that, PCA computes the vector which minimizes a function of the co-variance matrix. This vector is the principal eigenvector[3] of the co-variance matrix. The other (orthogonal) eigenvectors capture the residual variance and should represent noise. In our context, the use of PCA and then of a minimum variance-based criterion has the advantage of explaining most of the variance with a small number of eigenvectors. In this way, a few factors can be used for explaining, for example, user behavior. It is useful noting that there are other methods than PCA for representing factors — these methods are classified as decompositions [20].

As the eigenvectors of the co-variance matrix are a representation of the hidden factors, it is natural to be curious about the degree to which a document is affected by a factor. Therefore, the focus is on how the eigenvectors are used for *ranking* documents. If the objects are described by the $\mathbf{x}_i$'s, ranking in context reorders the vectors by the square of the projection between them and the eigenvectors $\mathbf{u}_j$'s which describe the contextual factors. Therefore, the ranking function is

$$|\mathbf{x}_i \cdot \mathbf{u}_j|^2 \tag{1}$$

where $|\mathbf{x}_i| = \sqrt{\sum_j x_{ij}^2} = 1$.

It is interesting to note that the formula resembles the inner product used in the VSM for IR and that PCA was already used in Latent Semantic Analysis for extracting hidden concepts from documents. However, the details of this ranking function can be uncovered and an explanation of why it is proposed can be obtained as reported in [19, 21].

In the following, we describe how to implement an IRF algorithm that captures the contextual factors. As an example, suppose the following six feature (column) vectors have been observed after seeing six (row) documents:

---

[2] When using PCA, co-variance matrix is suggested.

[3] The principal eigenvector is associated to the largest eigenvalue, which is a measure of the variance explained.

$$\mathbf{X} = \begin{bmatrix} -1.17 & -2.17 & -3.17 & 0.50 & 0.67 & 1.33 \\ -0.17 & -2.17 & 2.83 & 0.50 & -0.33 & 0.33 \\ -0.17 & -2.17 & 0.83 & -0.50 & -1.33 & -0.67 \\ 0.83 & 1.83 & 1.83 & -0.50 & 1.67 & 1.33 \\ 1.83 & -1.17 & -3.17 & -0.50 & -0.33 & -0.67 \\ -1.17 & 5.83 & 0.83 & 0.50 & -0.33 & -1.67 \end{bmatrix}$$

where the columns corresponds to, say, (1) display time, (2) scrolling, (3) saving, (4) bookmarking, (5) access frequency and (6) Web-page depth[4], respectively — all of these values may refer, for example, to time or frequencies, and can be seen as features of user behavior.[5] The following eigenvectors are then computed:

$$\mathbf{U} = \begin{bmatrix} -0.09 & 0.02 & 0.08 & -0.91 & -0.18 & -0.34 \\ 0.91 & 0.37 & 0.10 & -0.05 & -0.17 & 0.04 \\ 0.38 & -0.92 & -0.01 & -0.06 & 0.04 & -0.02 \\ 0.03 & 0.00 & -0.01 & 0.37 & -0.15 & -0.92 \\ 0.04 & 0.05 & 0.69 & -0.03 & 0.71 & -0.13 \\ -0.15 & -0.11 & 0.71 & 0.15 & -0.64 & 0.15 \end{bmatrix}$$

The values of an eigenvector are scalars between -1 and $+1$; the further a value is from 0 the more important it is. In this circumstance, "important" means that the feature to which the value corresponds is a significant descriptor of the contextual factor represented by the eigenvector. The value can be likened to an index term weight. As the values may be negative, the sign can express the contrast between features and then the presence of subgroups of features in the same contextual factor. For example, the first eigenvector, $\mathbf{u}_1$, tells that saving and bookmarking are least important, while the most important feature is scrolling.

Let $\mathbf{u}_j$ be one of these eigenvectors and $\mathbf{x}_i$ be an unseen document. The function of the distance between the document vector and the subspace spanned by the eigenvector is then used as a measure of the distance between the document and the contextual factor. Therefore, $\mathbf{x}_i \cdot \mathbf{u}_j$ is computed. If the unseen document vector is, say, $\mathbf{x}_i = (0.71, 0, 0, 0, 0.71, 0)$, then the distance is 0.03.

In the next section we describe an experiment that compared an IRF algorithm based on PCA that represents each feature separately with a comparator algorithm that uses a single centroid of all features.

3 Implicit Feedback Experiments

The aim of the experiment was to assess the retrieval effectiveness of an IRF algorithm that used the features of user behavior as feedback and translated this feedback into document rankings computed by Equation 1.

---

[4] The depth of a Web page is the number of links from the root of the Web site to the Web page itself.

[5] This example is inspired by the data set used in D. Kelly. *Understanding Implicit Feedback And Document Preference: A Naturalistic User Study.* PhD thesis, Rutgers, The State University of New Jersey, 2004.

3.1 Methodology

The interaction logs of real subjects were used to simulate a user who accesses a series of Web pages, spends time to read them, scrolls the browser window, moves the mouse and presses keyboard keys. The IRF algorithms under investigation are assumed to be part of a system that monitors user behavior and uses these interaction data as a source of IRF to retrieve and order the unseen documents. When the subject is known, the system records the data by user and then retrieves and ranks the unseen documents for the given user. The details of the simulation are as follows (let us name this algorithm EIG since it is based on the eigenvectors of a feature co-variance matrix):

1. The features of n documents seen by the user are observed and used for computing a representation of context by computing the contextual factors as follows:
   (a) the feature co-variance matrix is computed,
   (b) the eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ are extracted from the co-variance matrix — an eigenvector represents a contextual factor.
2. The documents unseen by the subject are ranked by Equation 1 for each eigenvector $\mathbf{u}_i$.

Multi-level usefulness scores assigned to documents by users have been used as ground truth information for evaluating this IRF algorithm and has not been used for computing the eigenvectors. Normalized Discounted Cumulative Gain (NDCG) [22] was used as a measure of retrieval effectiveness that was able to handle usefulness scores ranging in a non-binary scale.[6] NDCG is a performance metric that is able to make better use of multi-level judgments than precision, which generally must use binary relevance values. NDCG is a measure of distance between two rankings — the ranking produced by an experiment and the best ranking the experiment might produce.

For comparison purposes, the unique centroid vector of the cluster of n vectors of the documents seen by the user was computed. The inner product between the centroid vector and the unseen document vectors is then computed for ranking the unseen documents. Note that no document clustering is performed. Let us name this algorithm CTR. CTR was chosen because it exploits the same data used by EIG but aggregates all interaction feature vectors into a single factor, allowing us to determine the value of utilizing multiple factors, as permitted by EIG.

3.2 Document Features

The data set used in this experiment was gathered during the investigation of the Curious Browser reported in [23]. The set collects the data about 2,127 documents seen by 77 subjects and has information about the actions performed by the subjects whilst conducting self-determined Web browsing tasks, that is,

[6] The discount factor was 2.

without predefined tasks assigned by the experimenter. The following document features of the data set were used in our study: the time spent on a page (`page`), the time spent for horizontal scrolling (`hscroll`), the time spent for vertical scrolling (`vscroll`), the number of scroll events (`#scroll`), the time spent for moving the mouse (`mouse`), the number of the mouse clicks (`#mouse`), the number of times hitting the up arrow key (`#upkey`), the number of times hitting the down arrow key (`#downkey`), the time spent holding the up arrow key (`upkey`), the time spent holding the down arrow key (`downkey`), the time spent holding the page up key (`pgup`), the number of times hitting the page up key (`#pgup`), the time spent holding the page down key (`pgdown`), the number of times hitting the page down key (`#pgdown`), the number of slashes of the visited URL (`urldepth`).[7]

In addition to these features, we also have explicit multi-level ratings assigned by participants based on their own assessment on the usefulness of the document for their browsing activity. These ratings could then be used in the assessment of algorithm performance in our study.

In the next section we present the findings of our study.

3.3 Results

The experiments sought to compare the two IRF algorithms and determine whether there was a contextual factor which orders the unseen documents more effectively than other factors. To this end, the comparison with CTR would allow us to determine whether this "special" eigenvector exists, since CTR computes a single centroid vector. To be precise, the question: "Is there an eigenvector for which EIG "beats" CTR?" will be answered. This "special" eigenvector would allows us to personalize IRF to each user.

In order to establish the role played by the eigenvectors, an analysis was conducted to compare the effectiveness of CTR with the effectiveness of EIG by varying the eigenvector. That is, one eigenvector was fixed at a time and the documents were ranked using the fixed eigenvector. We did this for each subject. Table 1 reports NDCG of CTR and NDCG of EIG. The values in the table are shown after the user had viewed two documents (i.e., $n = 2$). This value has been chosen because it is small enough for evaluating the capability of the simulated system to perform effectively even if the feedback is limited. The number of unseen and ranked documents was $N - n$ where N is the total number of documents seen by the subject in the data set. The eigenvector which achieved the highest average NDCG of EIG was selected over all the eigenvectors. The table reports the composition of the eigenvector for each subject thus making a clear description of the behavior of each subject when accessing the Web pages.

The results suggests that, an eigenvector for which EIG is more effective than CTR almost always exists. Moreover, `page` (i.e., the time spent on each

[7] `urldepth` was added by the authors and was not provided by the data set. The number of slashes has been used because it is a measure of Web-page quality and is an endorsement of the Web-page when the end user selects it. The number of slashes is also known as URL depth and is used for successfully retrieving entry Web pages, which are often preferred by the users when finding resources [24].

lane) is the most important feature of user behavior for every subject. However, the best eigenvector varies its shape depending on the subject. For example, subject 5's behavior is also determined by `mouse` (i.e., time spent moving the mouse). Moreover, some features tend to contrast others. For example, subject 74 spends long periods of time on pages when they seldom scroll, and vice versa. Although `page` describes a common aspect of the interaction of every user, it was clear that each subject had a slightly different interaction style when seeking information, and more than one aspect of this style is necessary to distinguish between subjects. The presence of `page` means that it is necessary for tailoring retrieval to every user, but it is not sufficient since other features are necessary for maximizing retrieval effectiveness. These results suggest that tailoring eigenvectors to users leads to improved performance over algorithms that do not use such an approach. This finding is important because it justifies the design of IRF algorithms that learn from an individual user's interaction and adapt themselves to that user.

Table 1: The composition of the most effective eigenvector for each subject. Feature subgroups corresponding to negative weighs are italicized.

Subject	NDCG		Best Eigenvector
	EIG	CTR	
1	0.883	0.170	`page` (0.843); `vscroll` (0.527); `mouse` (0.107);
2	0.833	0.573	`page` (0.871); `vscroll` (0.442); `mouse` (0.214);
3	0.930	0.491	`page` (0.997); `mouse` (0.078); `pgdown` (0.015);
4	0.907	0.965	`page` (0.894); `mouse` (0.449);
5	0.767	0.654	`page` (0.971); `mouse` (0.238);
6	0.770	0.929	`page` (0.895); `mouse` (0.446);
8	0.933	0.114	`page` (0.746); `mouse` (0.666);
9	0.844	0.804	`page` (0.999); `mouse` (0.023);
10	0.822	0.951	`page` (0.850); *mouse* (-0.53);
11	0.722	0.734	`page` (0.981); `mouse` (0.161); `vscroll` (0.107);
12	0.836	0.741	`page` (0.966); `mouse` (0.253); `vscroll` (0.062);
13	0.916	0.469	`page` (0.957); `mouse` (0.286); `vscroll` (0.051);
14	0.935	0.840	`page` (0.900); `vscroll` (0.386); *mouse* (-0.20);
15	0.873	0.725	`page` (0.995); *mouse* (-0.09);
17	0.738	0.863	`page` (0.915); `mouse` (0.403); `vscroll` (0.025);
19	0.889	0.788	`page` (0.994); *mouse* (-0.04); *pgdown* (-0.10);
20	0.861	0.434	`page` (0.897); `mouse` (0.442);
21	0.658	0.671	`page` (0.827); *mouse* (-0.56);
22	0.868	0.838	`page` (0.967); `mouse` (0.255);
23	0.903	0.501	`page` (0.822); *downkey* (-0.01); *vscroll* (-0.21); *mouse* (-0.53);
24	0.976	0.784	`page` (1.000);

Continued on next page

Table 1 – continued from previous page

Subject	NDCG		Best Eigenvector
	EIG	CTR	
25	0.840	0.827	page (0.991); downkey (0.054); *mouse* (-0.04); *vscroll* (-0.11);
26	0.888	0.777	page (0.950); vscroll (0.264); mouse (0.166);
27	0.751	0.920	page (0.990); mouse (0.138);
28	0.631	0.548	page (0.748); mouse (0.663);
30	0.801	0.519	page (0.999); mouse (0.015);
31	0.938	0.912	page (0.999); mouse (0.015);
32	0.715	0.253	page (0.999); mouse (0.035);
33	0.880	0.626	page (0.925); mouse (0.277); vscroll (0.260);
34	0.849	0.767	page (0.947); mouse (0.322);
35	0.981	0.810	page (0.920); vscroll (0.392); *mouse* (-0.02);
36	0.825	0.411	page (0.859); vscroll (0.473); *mouse* (-0.20);
37	0.892	0.832	page (0.950); mouse (0.304); vscroll (0.048); *pgdown* (-0.01); *upkey* (-0.02); *downkey* (-0.05);
38	0.878	0.930	page (0.994); vscroll (0.093); mouse (0.054); *pgup key* (-0.01); *pgdown* (-0.02);
39	1.000	1.000	page (0.822); downkey (0.383); upkey (0.352); mouse (0.233);
40	0.907	0.848	page (0.913); mouse (0.408);
41	1.000	0.581	page (0.951); mouse (0.310);
42	1.000	0.778	page (0.979); *vscroll* (-0.05); *mouse* (-0.19);
43	0.920	0.898	page (0.999); mouse (0.021);
44	0.981	0.310	page (0.933); mouse (0.355); vscroll (0.056);
45	0.914	0.562	page (0.884); mouse (0.353); vscroll (0.306); pgdown (0.018);
46	0.847	0.630	page (0.997); mouse (0.081);
47	0.893	0.324	page (0.987); *mouse* (-0.16);
48	1.000	1.000	page (0.953); mouse (0.304);
49	1.000	1.000	page (0.953); mouse (0.304);
50	0.961	0.631	page (0.959); mouse (0.283);
51	0.953	0.564	page (0.889); mouse (0.458); hscroll (0.028);
52	0.771	0.605	page (0.900); mouse (0.435);
53	0.892	0.477	page (0.981); mouse (0.100); *vscroll* (-0.17);
54	0.834	0.169	page (0.935); vscroll (0.260); mouse (0.242);
55	0.909	0.587	page (0.883); vscroll (0.052); *mouse* (-0.47);
56	0.946	0.783	page (0.960); vscroll (0.278); mouse (0.032);
57	0.962	0.803	page (0.999); *vscroll* (-0.02);
58	0.887	0.870	page (0.988); *mouse* (-0.15);
59	1.000	1.000	page (0.913); *pgup key* (-0.03); *downkey* (-0.41);
60	0.856	0.789	page (0.891); vscroll (0.323); mouse (0.319);
61	1.000	0.863	page (0.957); mouse (0.252); vscroll (0.140);

Continued on next page

Table 1 – continued from previous page

Subject	NDCG		Best Eigenvector
	EIG	CTR	
63	0.814	0.928	`page` (0.996); `mouse` (0.092);
64	0.920	0.878	`page` (0.932); *`mouse`* (-0.36);
65	1.000	0.995	`page` (0.958); `mouse` (0.248); `vscroll` (0.143);
66	0.901	0.199	`page` (0.868); `vscroll` (0.473); `mouse` (0.152);
67	0.860	0.949	`page` (0.856); `vscroll` (0.472); `mouse` (0.211);
68	0.875	0.944	`page` (0.897); `mouse` (0.443);
69	0.990	0.929	`page` (0.850); `mouse` (0.527);
71	0.958	0.863	`page` (0.976); `vscroll` (0.177); `mouse` (0.127);
72	0.989	0.984	`page` (0.999); `vscroll` (0.042); `mouse` (0.018);
73	0.915	0.884	`page` (0.939); `mouse` (0.345);
74	0.903	0.532	`page` (0.980); *`mouse`* (-0.01); *`vscroll`* (-0.20);
75	0.962	0.224	`page` (0.971); `mouse` (0.199); `vscroll` (0.133);
76	0.713	0.558	`page` (0.995); `mouse` (0.091); `vscroll` (0.030);
77	0.760	0.632	`page` (0.985); `mouse` (0.162); `downkey` (0.051);
Avg.	0.923	0.774	
StDev.	0.088	0.238	

4 Conclusions and Future Work

In this paper a statistical framework that utilizes multiple sources of evidence present in an interaction context has been presented to discover hidden contextual factors that can be used for personalization. The eigenvectors extracted from a feature co-variance matrix observed from interaction are used as representation of the hidden contextual factors. These representations have been compared with an alternative using rich interaction logs (and associated metadata such as relevance judgments) gathered during a user study. Our findings demonstrate the effectiveness of these representations. In particular, it was shown that implicit feedback could be effective when the representation of the contextual factors are personalized to the user. Future work will address the challenge of selecting the best eigenvector automatically.

References

1. G. Salton and M.J. McGill. *Introduction to modern Information Retrieval*. McGraw-Hill, New York, NY, 1983.
2. M. Beaulieu. Experiments with interfaces to support query expansion. *J. of IR*, 53(1):8–19, 1997.
3. D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

4. D. Kelly and N.J. Belkin. Display time as implicit feedback: understanding task effects. In *Proc. of SIGIR*, pages 377–384, New York, NY, USA, 2004. ACM Press.
5. R.W. White, I. Ruthven, and J.M. Jose. A study of factors affecting the utility of implicit relevance feedback. In *Proc. of SIGIR*, pages 35–42, New York, NY, USA, 2005. ACM Press.
6. T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. of SIGIR*, pages 154–161, New York, NY, USA, 2005. ACM Press.
7. S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM TOIS*, 23(2):147–168, 2005.
8. M. Morita and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proc. of SIGIR*, pages 272–281, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
9. J. Budzik and K. Hammond. Watson: Anticipating and contextualizing information needs. In *Meeting of the ASIS*, pages 727–740, Medford, NJ, 1999.
10. R.W. White, I. Ruthven, and J.M. Jose. Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In *Proc. of SIGIR*, pages 57–64, New York, NY, USA, 2002. ACM Press.
11. J. Kim, D.W. Oard, and K. Romanik. Using implicit feedback for user modeling in internet and intranet searching. Technical report, College of Library and Information Services, University of Maryland at College Park, 2000.
12. H. Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proc. of IJCAI*, pages 924–929, Montreal, Quebec, Canada, 1995. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.
13. T. Joachims, D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In *Proc. of IJCAI*, pages 770–777, San Francisco, CA, 1997. Morgan Kaufmann.
14. E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 256–265, Madison, WI, July 1998.
15. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in knowledge discovering and data mining*. The MIT Press, 1996.
16. M. Melucci. Can vector space bases model context? In *Proc. of Context-based IR workshop*, Paris, France, July 2005. CEUR.
17. M. Melucci. Context modeling and discovery using vector space bases. In *Proc. of CIKM*, pages 808–815, Bremen, Germany, November 2005. ACM Press.
18. M. Melucci. Ranking in context using vector spaces. In *Proc. of CIKM*, pages 477–478, Arlington, VA, USA, November 2006. ACM Press.
19. M. Melucci. Exploring a mechanics for context-aware information retrieval. In *Proc. of the AAAI Spring Symposium on Quantum Interaction*, Stanford, CA, USA, March 2007. AAAI Press.
20. R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
21. C.J. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, UK, 2004.
22. K. Jarvëlin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM TOIS*, 20(4):422–446, 2002.
23. M. Claypool, P. Le, M. Waseda, and D. Brown. Implicit interest indicators. In *Proc. of IUI*, pages 33–40, 2001.
24. W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proc. of SIGIR*, pages 27–34, Tampere, Finland, 2002.