Know the Right People?
Recommender Systems for Web 2.0

Stefan Siersdorfer
University of Sheffield, UK
Dept. of Information Studies
s.siersdorfer@sheffield.ac.uk

Sergej Sizov
University of Koblenz, Germany
Dept. of Computer Science
sizov@uni-koblenz.de

Paul Clough
University of Sheffield, UK
Dept. of Information Studies
p.d.clough@sheffield.ac.uk

Abstract

Web 2.0 applications like Flickr, YouTube, or Del.icio.us are increasingly popular online communities for creating, editing and sharing content. However, the rapid increase in size of online communities and the availability of large amounts of shared data make discovering relevant content and finding related users a difficult task. Web 2.0 applications provide a rich set of structures and annotations that can be mined for a variety of purposes. In this paper we propose a formal model to characterize users, items, and annotations in Web 2.0 environments. Based on this model we propose recommendation mechanisms using methods from social network analysis, collaborative filtering, and machine learning. Our objective is to construct collaborative recommender systems that predict the utility of items, users or groups based on the multi-dimensional social environment of a given user.

1 Introduction

1.1 Motivation

We have entered the "knowledge age" where economic power or wealth is based on ownership of knowledge [15] and the ability to utilize knowledge for the improvement of services or products. "Knowledge workers" already outnumber any other kind of worker in highly developed economic systems. Their work is characterized by complex human-centric processes [20] describing information intensive and complex tasks comprising of the creation, retrieval, digestion, filtering, and sharing of (large amounts of) knowledge.

Basically, the paradigm of a flexible environment that supports the user in producing, organizing, and browsing the knowledge is not new. It originates in the early 1940s, long time before the first personal computers and new communication tools like the internet became available. The conceptual design of Vannevar Bush's Memex [10] (an acronym for Memory Extender) is probably the most cited (e.g. [14]) and criticized (e.g. [8]) representative of the early conceptual work. In his article, Bush describes the integrated work environment that was electronically linked to a repository of microfilms and able to display stored content and automatically follow references from one document to another. A number of visionary ideas from this early conceptual work can be recognized in state-of-the art information systems (cross-references between documents, browsing, keyword-based annotation of documents using the personal "codebook", automatic generation of associative trails for content summarization, etc.).

However, an important difference attracts attention if we consider the aspect of collaboration. In fact, knowledge workers of today are never working isolated. The internet and the World Wide Web (WWW) provide an infrastructure on top of which a variety of communication channels and collaborative environments have been established. Collaboration is one of the major tasks of the knowledge worker as it denotes the exchange of information and transfer of knowledge. It is vital for any collaborative human work, e.g. for coordinating activities, reporting on work progress, discussing solutions and problems, or disseminating new information. Efficient collaboration infrastructure is probably one of the key differences of modern work environments in contrast to isolated Memex-style solutions from the past.

For this reason, it is not surprising that knowledge and data sharing in Web 2.0 applications has rapidly gained popularity. Despite disagreement on the exact definition of Web 2.0, it is common to find community and collaboration as key concepts in this latest online phenomenon. Increasingly, online content is being created, edited and shared by whole communities of users, demonstrated by the popularity of applications such as Flickr[1], YouTube[2] and Del.icio.us[3]. Web 2.0 applications provide a rich set of structures and annotations that can be mined for a variety of purposes. For example, Flickr postings are accompanied with a variety of descriptive metadata, such as creator (and/or owner), a textual description, thematic tags, temporal and geographic information and comments by other Flickr users on specific regions of uploaded pictures. Using these structures, a variety of relationships between users, tags, pictures, and groups can be explored.

Collaboration in Web 2.0 environments induces new problems and challenges. In many cases, the size of online communities has increased rapidly over the last decades and large amounts of shared data are now available. This makes the discovery of relevant content and finding users with shared interests a difficult enterprise. Ideally, the Web 2.0 platform should provide the user with adaptive browsing mechanisms and recommendations for potentially relevant content, users, or annotations. Unfortunately, recent platforms are rather limited in supporting self-organization. The vast majority just offers explicitly defined groups/topics of interest along with suitable subscription, dissemination, and communication mechanisms.

[1] http://www.flickr.com
[2] http://www.youtube.com
[3] http://del.icio.us/

Static groups, however, cannot properly reflect and support the evolution of user interests and one-day thematical hotspots, such as "Valentine's day" or "recent Oscar nominations". Therefore, it is necessary to have views that put together users and content by identifying their topics of interest and current demands "on the fly".

In this article, we address the following questions: What is the generalized model for recommender scenarios in Web 2.0 applications? How can we represent and utilize available meta information? How can we apply automatic techniques for making recommendations? In doing so, we introduce a formal notion of Web 2.0 relationships, formalize the recommender problem, discuss possible solutions and evaluation methodologies, and identify promising open questions and research directions for upcoming research.

1.2 Analysis of Requirements

Many algorithms and methods from the Web IR domain can be adapted (and have been adapted in the last decades) for applications that address Web 2.0 problems. However, some important differences of the collaborative environments should be taken into account:

- Analogously to many other collaborative environments, Web 2.0 applications exhibit the behavior of a social network. However, the underlying evolution behavior may substantially differ from existing models for the Web scenario (e.g. preferential attachment [4] and random rewiring [32]). Therefore, these models do not necessarily provide the best fit for Web 2.0 social network characteristics, such as network diameter, characteristic path length, or clustering coefficient. Moreover, there are multiple levels of system growth with potentially different evolution patterns, e.g. micro-behavior (particular users) and macro-behavior (user groups).
- The browsing/recommendation scenario substantially differs from more common IR methods used for Web retrieval. On one hand, there are multiple dimensions (users, user groups, tags, comments, personal favorite lists) that can be used for object characterization. On the other hand, particular dimensions (e.g. annotations of the given item picture or video) tend to be extremely sparse. A suitable hybrid model for multimedia, text, and user mining may be required for constructing integrated recommender applications.
- Due to a large number of available resources, recommender applications should provide a highly efficient infrastructure for data representation and personalization. Typically, the user is not interested in obtaining a vast number of recommendations; however, the acceptance of the recommender system is crucially dependent on the quality of best recommendations in the ranked list, and on the required response time for getting these recommendations.
- Almost all Web 2.0 applications are highly dynamic environments with frequent rates of change for both content and user interests. Themes of high interest (e.g. pictures associated with some event such as natural disasters, trips, or conferences) may have a short lifespan.

In order to address these custom properties of collaborative Web 2.0 applications, the target recommender system should satisfy the following top-level requirements:

Flickr Query	Cardinality of the unordered result set
www	3472
www 2007	2297
www banff	47
www conference	34
www 2007 banff	29
www conference 2007	19
www conference banff	8
www 2007 conference banff	8

Figure 1: Samples of Querying Flickr

1. It should take into account a specialized model of dependencies between users, items, and annotations that provides a good fit for observed properties of the folksonomy.
2. The multi-dimensional model should capture various aspects of the folksonomy, including application-specific ones (e.g. favorites lists, comments, user groups, etc.) and correlations between them.
3. The system should provide at least a limited number of high-quality results with short response times, e.g. using appropriate $top - k$ retrieval algorithms in the background.
4. The system should provide mechanisms for trend detection in folksonomies and support trend-based recommendation of new content to the custom user.
5. The system should provide mechanisms of personalization, e.g. by using a multi-dimensional user-specific context that captures the "small world" of the user's annotations and items, other related users, potentially relevant items and discussion threads with the user's participation.

1.3 Example Scenario

Let us consider an example scenario in which a user is looking for pictures about the WWW conference 2007 in Banff, Canada. By querying Flickr for "WWW" or "WWW Banff", he would obtain an unordered results list that contains between 2,000 and 3,000 matches (Figure 1). Since the system cannot provide a meaningful ranking for these results, finding relevant items becomes a hard needle-in-a-haystack search problem. Of course, the user could refine the conjunctive query by experimenting with more specific query formulations, e.g. by incrementally adding more and more search keywords (and thus increasing restrictivity). However, in practice, this leads to a rapid decrease in recall: the query "WWW Conference Banff 2007" returns only a few potentially relevant matches. In both described cases, finding a suitable number of potentially interesting pictures remains a difficult task.

It is clear that a suitable recommender system should aim to provide a better ratio between topic restriction and cardinality of the results set. For example, by analyzing a user's personal profile (e.g. favorites list, participation in groups and comments) the system could suggest a community of other users with similar interests or professional background (e.g. other computer scientists that participated in the WWW 2007 conference). In the next step, the query "WWW 2007" could be executed with restricted scope on annotated items provided by these related users. Alternatively, the query could be expanded by search terms often used by the community and highly correlated with the

Figure 2: Clusters of Related Keywords in Folksonomies

user's search keywords (e.g. by identifying a high correlation between the tags "WWW" and "WorldWideWeb"). To a certain extent, this functionality is supported by systems like Flickr, Bibsonomy and or del.icio.us; however, clusters of related keywords are estimated in a global setting and do not capture the personal context of a particular user (Figure 2).

Furthermore, by analyzing evolving interests of the community, the system should be able to gather items from related events (e.g. pictures from the follow-up conference WSDM 2008, announced on the WWW 2007 website), and present corresponding matches as new recommendations to the user. In the optimal case, the recommender algorithms should run as background processes without the need for human intervention or relevance feedback.

1.4 Related Work

Schmitz et al. have formalized folksonomies and discuss the use of association rule mining for analyzing and structuring them in [28]. The recent work on folksonomy-based web collaboration systems includes [12], [16], and [23] which provide good overviews of social bookmarking tools with special emphasis on folksonomies, and [**?**] which discusses strengths and limitations of folksonomies in a more general setting. In [**?**], a model of semantic-social networks for extracting lightweight ontologies from del.icio.us is defined.

The analysis of topological properties is well-known in the areas of complex networks [26; 25; 2; 11; 6] and social network analysis (SNA). Typical examples of such measures are the clustering coefficient and the characteristic path length in the tripartite undirected hypergraph: $G = (V, E)$, where $V = U \dot{\cup} T \dot{\cup} R$ is the set of nodes, and $E = \{\{u, t, r\} \mid (u, t, r) \in Y\}$ is the set of hyperedges. In other words, the hypergraph captures relationships between users, annotations, and items. An equivalent common view on folksonomy data is a quadruple $\mathbb{F} := (U, T, R, Y)$. This structure is known in Formal Concept Analysis [33; 13] as a *triadic context* [22; 29].

There are several systems working on top of del.icio.us to explore the underlying folksonomy. CollaborativeRank[4] provides ranked search results on top of del.icio.us bookmarks. The ranking takes into account how early someone bookmarked an URL and how many people followed them. Other systems show popular sites (Populicious[5]) or focus on graphical representations (Cloudalicious[6], Grafolicious[7]) of statistics about del.icio.us.

In many cases, suitable recommendations can be obtained by analyzing link-based authority measures of the folksonomy. Site ranking algorithms, for instance the PageRank algorithm [7], use topological information embedded in a directed network to infer the relative importance of nodes. Analogously, a node ranking procedure for folksonomies, the FolkRank algorithm, has been introduced in [18]. In contrast with PageRank, FolkRank also provides useful information in the case of undirected networks. Taking on a different perspective, community detection algorithms are able to detect relation similarities at a higher level. A yet different procedure is the Markov Clustering algorithm (MCL) in which a renormalization-like scheme is used in order to detect communities of nodes in weighted networks [31].

Detection of global trends in the community is an additional valuable source of information for constructing recommendations. In [3], the evolution of the relationship graphs over time is analyzed. The application of the proposed method lies in the improved detection of current real-life trends in search engines. In the future, new search methods for folksonomies should support adaptation of [3] to the Web 2.0 scenario.

Kleinberg [19] summarizes several different approaches to analyze online information streams over time. He distinguishes between three methods to detect trends: using normalized absolute change, relative change and a probabilistic model. The popularity gradient is related to the second approach, but differs insofar as it allows for the discovery of *topic-specific* trends and honours steep rises more if they occur higher in the ranking. The text mining scenario described in [19] requires focusing on words that are neither too frequent nor too infrequent.

Common recommender systems are usually used in one of two contexts: (1) to help users locate items of interest they have not previously encountered, and (2) to judge the degree of interest a user will have in item they have not yet rated. With the growing popularity of on-line shopping, E-commerce recommender systems have matured into a fundamental technology to support the dissemination of goods and services [27]. Much research has been undertaken to classify different recommendation strategies [9; 17], but here we divide them broadly into two categories: *Collaborative* and *Content-based* recommendations.

Collaborative recommendation is probably the most widely used and extensively studied technique that is founded on one simple premise: if user A is interested in items w, x, and y, and user B is interested in items w, x, y, and z, then it is likely that user A will also be interested in item z. In a collaborative recommender system, the ratings a user assigns to items is used to measure their commonality with other users who have also rated the same items. The degree of interest for an unseen item can be deduced for a particular user by examining the ratings of their neighbors. It has been recognized that a user's interest may change over time, so time-based discounting methods have been developed [5] to reflect changing interests.

Content-based recommendation represents the culmination of efforts by the information retrieval and knowledge

[4] http://collabrank.org/

[5] http://populicio.us/

[6] http://cloudalicio.us/

[7] http://www.neuroticweb.com/recursos/del.icio.us-graphs/

representation communities. A set of attributes for the items in a system is determined, such as terms and their frequencies for documents in a repository, so the system can build a profile for each user based on the attributes present in the items that a user has rated highly. The interest a user will have in an unrated item can then be deduced by calculating its similarity to their profile based on the attributes assigned to the item.

Such systems are not without their deficiencies, the most prominent of which arise when new items and new users are added to the system - commonly referred to as the *ramp-up* problem [21]. Since both content-based and collaborative recommender systems rely on ratings to build a user's profile of interest, new users with no ratings have neutral profiles. When new items are added to a collaborative recommender system, they will not be recommended until some users have rated them. Collaborative systems also depend on the overlap in ratings across users and perform badly when ratings are sparse (i.e. few users have rated the same items) because it is hard to find similar neighbors.

Hybrid recommender systems, i.e. those which make use of collaborative and content based approaches, have been developed to overcome some of these problems. For example, collaborative recommender systems do not perform well with respect to items that have not been rated, but content-based methods can be used to understand their relationship to other items. Hence, a mixture of the two approaches can be used to provide more robust systems. More recent recommender systems have also investigated the use of ontologies to represent user profiles [24]. Benefits of this approach are a more intuitive profile visualization and the discovery of interests through inferencing mechanisms.

To the best of our knowledge, this paper is the first to describe the application of recommender systems on Web 2.0 structures.

2 Recommender Systems for Web 2.0

2.1 Objects and their Relationships

The structure of a collaborative sharing framework can be considered as a tripartite network with ternary relations (assigned tags) between users $u \in U$, resources (e.g. images, media files) $r \in R$ and associated tags (in our case arbitrary text labels) $t \in T$. The set of all relations of the framework is therefore $Y \subseteq U \times T \times R$ [28]. An equivalent representation of the framework is the hypergraph $G := (V, E)$ with vertices $V = U \cup T \cup R$ and hyperedges $E \subseteq Y$.

2.2 Information Clouds

The natural way to characterize elements from U and R in a sharing framework is a collection of element-specific tags $t_i \in T, i = 1..k$. We generally define an information cloud as the tuple

$$\mathcal{T} := (Y^*, f) \tag{1}$$

where $Y^* \subseteq Y$ is a context-dependent subset of all tag relations and

$$f(t) : Y^* \rightarrow [0..1] \tag{2}$$

a *tag-rank function* that computes a score for each $t \in T^* \subseteq Y^*$.

The score can be interpreted e.g. as a relevance measure of the given tag with respect to the characterized subject (e.g. $0 =$ least relevant, $1 =$ most relevant).

The introduced notion of clouds can be directly used to characterize elements from U and R. For example, the *user-centric* cloud $\mathcal{T}_u$ (i.e. user-specific collection of tags for the user $u \in U$) and a *resource-centric* cloud $\mathcal{T}_r$ (i.e. all associated tags of a single resource $r \in R$) can be expressed as

$$\mathcal{T}_u := (Y_u, f), \quad Y_u \subseteq u \times T \times R, \tag{3}$$
$$\mathcal{T}_r := (Y_r, f), \quad Y_u \subseteq U \times T \times r, \tag{4}$$

Analogously, more complex *community-centric* clouds can be used to summarize all tags used by a group of users $U^* \subseteq U$

$$\mathcal{T}_{U^*} := (Y_{U^*}, f), \quad Y_{U^*} \subseteq U^* \times T \times R, \tag{5}$$

or for a collection of resources $R^* \subseteq R$

$$\mathcal{T}_{R^*} := (Y_{R^*}, f), \quad Y_{R^*} \subseteq U \times T \times R^*, \tag{6}$$

Finally, a combination of (5) and (6) results in a community cloud about users and resources

$$\mathcal{T}_{U^*R^*} := (Y_{U^*R^*}, f), \quad Y_{U^*R^*} \subseteq U^* \times T \times R^*, \tag{7}$$

2.3 Information Tensor

The cloud notion can be used to capture pairwise dependencies between Web 2.0 attributes like users and tags. In a more general setting, the dependencies between users, items, tags, and further interesting entities (e.g. relationships by comments, users participating in groups or personal favorites lists) can be represented as a multi-dimensional tensor. Formally, we write an Mth order tensor as $\chi \in \mathbb{R}^{N_1 \times .. \times N_M}$, where $N_i, i = 1..M$ is the dimensionality of ith aspect. We notice that matrix unfolding of χ (i.e. vectors in $\mathbb{R}^{N_d}$ obtained by keeping index d fixed and varying the other indices) correspond to the information clouds introduced in the previous section.

For dynamic evolution scenarios, a sequence of information tensors $\chi_1..\chi_n$ can be considered. Each tensor χ_i corresponds to the snapshot of the Web 2.0 framework at a different timepoint (e.g. by monitoring system activities over time). If n is an integer that increases with time, the sequence is called a tensor stream [30]. Both sequences and streams allow for analysis of dynamic patterns in the corresponding Web 2.0 community. For instance, methods like DTA [30] have been especially designed for finding highly correlated dimensions and monitoring them, detection of anomalies, and clustering.

By choosing suitable projections of χ, we can easily restrict the scope to particular dimensions and map the generalized model notion onto existing problem definitions from the area of recommender systems that will be discussed in the next section.

3 Proposed Recommender System Design

In this section, we will show how concepts from recommender systems can be applied to Web 2.0 applications. We will concentrate on Flickr as a prominent example; however, the proposed techniques could also be applied to any other Web 2.0 scenario. Given a large data set, the objective of a recommender system is to propose a subset of items from this data set to a user that are potentially relevant or 'interesting'. For example, in Flickr these items can be photos, groups, or other users. This leads to recommendations such as:

- Given a user, recommend photos which may be of interest.
- Given a user, recommend users they may like to contact.
- Given a user, recommend groups they may like to join.

In the remainder of this section we will first provide a formal notion of recommender systems and show how this can be applied to a scenario such as Flickr. We then discuss three approaches to tackle the recommender problem: content-based methods, collaborative methods, and hybrid methods. We will use notions based on a recent survey on recommender systems [1].

3.1 Formalizing the Problem

Consider a utility function

$$ut : U \times S \rightarrow R \tag{8}$$

, where U is a set of users, S a set of items, and R a set of relevance values (e.g. real values in $[0, 1]$).

The objective of a recommender system is to choose for a user $u \in U$ an item $s'_u \in S$ that maximizes the user's utility:

$$s'_u = argmax_{s \in S} ut(u, s) \tag{9}$$

Usually, the utility function is defined over a subspace of $U \times S$. This requires that ut must be extrapolated to the whole space $U \times S$.

In the simplest case for Flickr, U corresponds to the set of Flickr users. There are extensions and generalizations possible: U could alternatively consist of tuples $(user, photo)$, meaning a user viewing (or commenting on) a photo is provided with a list of other related photos. Since the result of this recommendation depends upon the photos, as well as the user's profile, this is an example of "personalization".

The set of items S could correspond to the set of photos (likely the most obvious option), the set of other Flickr users, the set of s groups, or tags in Flickr.

The most interesting problem is the computation of relevance values R. For classical recommender systems, relevance directly assigned by the users is available, typically in the form of a star-rating. For example, in the movie application MovieLens.org, users assign 0 stars as the lowest rating and 5 stars as the highest. In Flickr, and many other Web 2.0 applications, a direct rating is not available[8]. However, annotations supplied by users can be considered as *implicit ratings*.

For a photo the following information can be used:

- The photo belongs to the user. In this simple case we might assume that the user is interested in the photos that he has uploaded. To obtain a more fine-grained measure, the length of the textual description of the photo and the number of tags could be taken into account (the intuition behind this is that users will put more effort into the annotation of photos that are interesting to them).
- The user has marked the photo as a favorite. This is probably the most direct positive relevance assignment possible in Flickr.
- The user writes one or more comments about the photo. This implies that for the user, it was worth the effort of making a statement about the photo (whether positive or negative). More enhanced methods could take the length and date of the comment into account, and use sentiment classification to categorize the comment as positive or negative.

For assigning relevance to other users, the following clues can be considered:

- A user is on the contact list of another user. In this case, it is likely that both users share similar interests.
- A user has saved photos from another user as their favorites.
- A user has written comments on another user's photos.

Similar relevance clues can be distinguished for other items such as groups or tags.

An overall relevance function can combine these annotations (e.g. by using a weighted linear combination of evidences). It should be noted that in the way previously described, we obtain just relevance values for a subset of items already known to the respective users. In the subsequent paragraphs, we will show how we can extrapolate this and other information to recommend new items to the user.

3.2 Content-based methods

For content-based methods, the user will be recommended items similar to those preferred in the past. The simplest, and most direct approach, is to estimate the utility $ut(u, s)$ of item s for user u based on the utilities $ut(u, s_i)$ assigned by user u to items s_i that are 'similar' to s. Formally, given a content representation $Content(s)$ and a content-based profile $ContentBasedProfile(u)$ of a user u, the utility function is usually defined as:

$$ut(u, s) = score(ContentBasedProfile(u), Content(s)) \tag{10}$$

where the $score$ function should produce high relevance values if $ContentBasedProfile(u)$ is related to $Content(s)$.

In Flickr, one approach is to represent both content and user profile as feature vectors (e.g. using a classical IR vector-space model). In this approach, the features can be weighted to vary their "importance". A common term-weighting strategy in IR is known as $tf * idf$ where tf denotes the frequency of a term and idf the inverse document frequency (i.e. the number of documents a term appears in). The intution behind this weighting is that a term is considered more important (i.e. more discriminative) if it occurs more frequently in fewer documents.

Using this approach, in Flickr photos can be represented as:

- Computing a $tf * idf$ vector from the set of tags associated with the photo.
- Computing a $tf * idf$ vector from the textual description of the photo.
- Computing a group vector with each dimension corresponding to the group the photo belongs to.

Possible representations of a user-profile can be obtained by:

- Computing a $tf * idf$ vector of the user description in his profile

[8] For YouTube a star-rating is available for the videos but not for other items such as users, groups and tags

- Computing the average of the user's photos vectors
- Computing a vector representing the groups of the user
- Computing the average of the photos from other users assigned as favorites or commented by this user

The first two options for describing user content can be used for the content-based profile as well as for the content of the user (if itemset S consists of users).

Given a vector representation $\vec{u}$ of $ContentBasedProfile(u)$ and $\vec{s}$ of $Content(s)$, the cosine measure can be used as a $scoring$ function (or similarity measure) to obtain:

$$ut(u,s) = cos(\vec{u},\vec{s}) = \frac{\vec{u}\cdot\vec{s}}{||\vec{u}||\cdot||\vec{s}||} \tag{11}$$

Machine Learning Approach Alternatively, relevance assignment can be formulated as a machine learning problem: given a set of items S_{pos} (represented as feature vectors as described above) relevant to the user, and S_{neg} that are not relevant to the user, train a binary classifier (with the two classes "relevant for the user" and "not relevant for the user") on these instances. Based on the trained model, it is then possible to estimate the relevance of new items. For Flickr, S_{pos} can be obtained using the user annotations (favorites, comments, contacts, etc.) as described in Section 3.1.

For example, linear support vector machines (SVMs) construct a hyperplane $\vec{w}\cdot\vec{x}+b=0$ separating the set of positive training examples from a set of negative examples with maximum margin δ. For a new previously unseen, item $\vec{d}$, the SVM simply tests whether the item lays on the "positive" side or the "negative" side of the separating hyperplane. In addition, the distances of the test items from the hyperplane can be interpreted as classification confidences.

3.3 Collaborative Methods

In *collaborative recommender systems*, also coined *collaborative filtering systems*, the user is recommended items that people with similar preferences have liked in the past. Formally, the utility $ut(u,s)$ of item s and user u is estimated based on the utilities $ut(u_j,s)$ assigned to item s by those users $u_j \in U$ who are similar to user u. The value of an unknown rating $ut(u,s)$ is usually computed as an aggregate of the ratings of other users (e.g. the N most similar) for item s:

$$ut(u,s) = aggr_{u'\in U'} ut(u',s) \tag{12}$$

, where U' is the set of N users most similar to u. Examples for aggregations given in [1] are average sum or weighted sum (weighted by the user similarities).

In section 3.2 we described several ways to obtain vector respresentations $\vec{u}$ of users u. Using a similarity measure such as the cosine measure for pairs of users, we can compute the N most similar users. The relevance assignment $ut(u',s)$ can be obtained using implicit ratings of other users described in Section 3.1.

Alternatively, we can take information from the social networks implicitly contained in Flickr into account to find similar or related users. Formally, these networks can be described, for instance, by the following graphs:

- *Contact graph* $G_{contact}(U,V)$ with $(u_1,u_2)\in V$ iff user u_2 is in the contact list of user u_1.
- *Comment graph* $G_{comment}(U,V)$ with $(u_1,u_2)\in V$ iff user u_1 has written a comment on a photo of user u_2.
- *Favorites graph* $G_{favorites}(U,V)$ with $(u_1,u_2)\in V$ iff user u_1 has assigned a photo of user u_2 as favorite.
- *Group graph* $G_{group}(U,V)$ with $(u_1,u_2)\in V$ iff user u_1 and user u_2 are are members of the same group.

Possible extensions are weighted graphs, taking e.g. the number of comments or favorites in $G_{comment}$ or $G_{favorites}$ into account or normalizing the weights in $G_{contact}$ by the overall number of contacts. Furthermore, we can consider the combination of graphs, computing, e.g. the union of edge-sets of distinct graphs.

We can find related users by traversing the social network graphs. For a user u we can, e.g., consider all users that are connected by a path of length $\leq k$, where k is parameter to be determined. How to tune the parameters, which graphs to choose for the search and how to combine the results are open research questions.

3.4 Hybrid Methods

For recommender systems, the sparsity of the data can be a serious problem [1]. For instance, in the collaborative approach described in the previous Section 3.3, for many items, there might be no implicit user-feedback available from the set of similar users. To overcome this problem, hybrid methods can be applied by incorporating content-based relevance assignments to obtain utility values $ut(u',s)$. Alternatively, collaborative and content-based methods can be first run separately, and then their predictions combined. For a more exhaustive review of hybrid methods see [1].

4 Evaluation Strategies

In the previous section, we have proposed various methods for representing objects in folksonomies, using annotations and implicit information, and recommender system design. To show the viability of our approaches, besides a deeper theoretical analysis and a statistical analysis of the data, a thorough experimental evaluation of quality of recommendations is necessary.

Evaluating recommendations in Web 2.0 applications is a difficult task for several reasons. First, the absence of established reference datasets with large amounts of manually verified and labeled recommendations may require comprehensive user studies with relevance feedback. This makes reliable and reproducible large-scale evaluation very hard and time-consuming. Second, there is a significant challenge in deciding what combination of measures should better characterize the recommender quality in a comparative evaluation. Ideally, the evaluation should be objective in reflecting the quality of recommendations with respect to realistic user needs (and be orthogonal to the functionality of the underlying method). For instance, in our previously introduced application scenario, we may measure the ability of the algorithm to reconstruct "hidden" annotations of pictures (i.e. existing annotations that have been removed before applying query expansion). In this case, we would confirm the basic functionality of the proposed method, which however cannot be considered as objective proof of recommender quality.

In this section we describe two possible evaluation approaches: evaluation based on user studies and evaluation

using implicit additional user information that can be directly inferred from Web 2.0 sources.

4.1 Manual Evaluation and User studies

High recommendation accuracy alone does not necessarily provide users with an effective and satisfying experience. A good recommender framework should also provide good usefulness [17]. A system that always recommends only highly popular items is probably not helpful in finding interesting "hidden" dependencies between communities, especially for a Web 2.0 environment. For example, a recommender system might tend to suggest conference pictures from other participants of to a WWW 2007 participant. Basically, this recommendation is highly accurate but rather obvious; recommended matches can be accessed by the user directly (e.g. by visiting the "WWW 2007" user group) without recommender assistance. Much more valuable would be a recommendation of images from press releases, photos from associated events like PhD workshops, or pictures of the Banff city center that are rather weakly related to the core WWW 2007 conference. Therefore, we ideally need new dimensions for analyzing recommender systems that consider both the "correctness" and the "non-obviousness" of the recommendations. Suitable dimensions are novelty and serendipity, which have been previously addressed in IR literature.

The subjective evaluation of "correctness" and "novelty" can be achieved through systematic user studies with aposteriori verification. In this evaluation scenario, recommendations of new items can be presented to multiple real users with different profiles of interest. The top-k part of the recommendation list (e.g. $k = 10$) is fully evaluated by each user by assigning scores for aspects such as "correct" and "interesting" (e.g. integer values between $1..5$).

Conceptually, manual inspection of the result set would provide the best evidence for evaluation. However, this requires comprehensive human experiments, and thus is often not scalable for large Web 2.0 platforms like Flickr or Del.icio.us.

4.2 Using Implicit Information for Evaluation

An alternative approach to approximate IR-style quality measures is the apriori method with an (estimated) gold standard. Metrics such as accuracy can be constructed by predicting the k items for which the relevance (or irrelevance) is known. A suitable approximation could be achieved by using individual favorite lists and comments, which can be considered as an indication of relevance. The recommender method should be constructed in such a way that these dimensions remain 'invisible' for the recommendation model; in other words, these dimensions must be artificially removed from the user-specific information tensor χ. An alternative is to keep these dimensions for a training set and evaluate the recommender system on a disjoint test set. The ability of the method to reconstruct favorite/comment lists (measured by the overlap between the top-k recommended items and the user-specific collection of such references, plus normalization in order to make estimates for different users comparable) can be treated as an accuracy measure.

A drawback of this methodology, however, is the absence of *negative* test samples. In fact, by using explicitly given comments and favorites we indirectly claim *all* remaining items as irrelevant, which is not entirely correct. Moreover, in practice (and in the manual evaluation scenario explained above) we are interested in finding *additional* relevant items beyond known favorites and/or comments. In a better experimental setting, the recommender method could be provided with a set of explicitly known "positive" and "negative" samples, whereby the "negative" collection could be collected through additional user studies. The ability of the method to correctly recognize "true positives" and to prioritize them in the top-k result set may provide a better accuracy estimate.

To this end, we assume that a combination of both manual user assignments and use of implicit information gathered from the folksonomy will provide the most comprehensive quality estimate.

5 Conclusion and Future Work

In this paper, we have discussed a design methodology for recommender systems in Web 2.0 applications. We have stated specific top-level requirements for recommender systems and ways of addressing them. The core representational model of our methodology captures multi-dimensional dependencies between users, items, and annotations in form of a multi-dimensional tensor. By choosing suitable projections we restrict the scope to particular dimensions of interest and can map the tasks to existing problem definitions from the area of recommender systems.

An important building block in the system design is the evaluation methodology. In this paper, we discussed pros and contras of possible evaluation strategies (apriori/aposteriori evaluation, manual result inspection vs. automated IR-style measurements) and identified the advantages of an integrated approach.

The results presented here can be summarized as a preliminary system design for Web 2.0 recommender infrastructures that will be refined and systematically evaluated in our future work. Our long-term objective is the design of scalable and reliable assistance methods that individually guide particular users through large-scale multi-dimensional Web 2.0 frameworks towards promising search results.

Acknowledgements

This work has been partially supported by the European project "Semiotic Dynamics in Online Social Communities" (Tagora, FP6-2005-34721) and the EU-funded project Memoir.

References

[1] G. Adomavicius and A. Tuzhilin. Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, 2005.

[2] R. Albert and A. Barabasi. Statistical mechanics of complex networks. *Review of Modern Physics*, 74:47–97, 2001.

[3] Einat Amitay, David Carmel, Michael Herscovici, Ronny Lempel, and Aya Soffer. Trend detection through temporal link analysis. *J. Am. Soc. Inf. Sci. Technol.*, 55(14):1270–1281, 2004.

[4] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509, 1999.

[5] D. Billsus and M. Pazzani. User Modeling for Adaptive News Access. 10(2-3):147–180, 2000.

[6] K. Borner, Soma Sanyal, and A. Vespignani. Network science: a theoretical and practical framework. *Annual Review of Information Science and Technology*, 41:537–607, 2007.

[7] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[8] M. Buckland. Emmanuel Goldberg, Electronic Document Retrieval, and Vannevar Bush's Memex. *JASIS*, 43(4):284–294, 1992.

[9] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[10] V. Bush. As We May Think. *The Atlantic Monthly*, 176(1):101 – 108, 1945.

[11] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of Networks: From Biological Nets to the Internet and WWW (Physics)*. Oxford University Press, Inc., New York, NY, USA, 2003.

[12] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. In *Proc. 15th Int. WWW Conference*, May 2006.

[13] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical foundations*. Springer, 1999.

[14] J. Gemmell, G. Bell, and R. Lueder. MyLifeBits: a personal database for everything. *Commun. ACM*, 49(1):88–95, 2006.

[15] Stephen Haag, Maeve Cummings, and Donald J. McCubbrey. *Management information systems for the information age*. Irwin McGraw-Hill, 3 edition, 2002.

[16] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine*, 11(4), April 2005.

[17] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[18] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information Retrieval in Folksonomies: Search and Ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, June 2006. Springer.

[19] J. Kleinberg. Temporal dynamics of on-line information streams. In M. Garofalakis, J. Gehrke, and R. Rastogi, editors, *Data Stream Management: Processing High-Speed Data Streams*. Springer, 2006.

[20] S. L. Kogan and M. J. Muller. Ethnographic study of collaborative knowledge work. *IBM Systems Journal*, 45(4):759, 2006.

[21] J. A. Konstan, J. Reidl, A. Borchers, and J.L. Herlocker. Recommender systems: A grouplens perspective. In *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08)*, pages 60–64. AAAI Press, 1998.

[22] F. Lehmann and R. Wille. A triadic approach to formal concept analysis. In G. Ellis, R. Levinson, W. Rich, and J. F. Sowa, editors, *Conceptual Structures: Applications, Implementation and Theory*, volume 954 of *Lecture Notes in Computer Science*. Springer, 1995.

[23] B. Lund, T. Hammond, M. Flack, and T. Hannay. Social Bookmarking Tools (II): A Case Study - Connotea. *D-Lib Magazine*, 11(4), 2005.

[24] S. Middleton, N. Shadbolt, and D. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.

[25] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167, 2003.

[26] Romualdo Pastor-Satorras and Alessandro Vespignani. *Evolution and Structure of the Internet: A Statistical Physics Approach*. Cambridge University Press, New York, NY, USA, 2004.

[27] Ben J. Schafer, Joseph A. Konstan, and John Riedi. Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166, 1999.

[28] C. Schmitz, A. Hotho, R. Jaeschke, and G. Stumme. Mining Association Rules in Folksonomies. pages 261–270, 2006.

[29] Gerd Stumme. A finite state model for on-line analytical processing in triadic contexts. In *ICFCA*, pages 315–328, 2005.

[30] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Philadelphia,USA*, pages 374–383, 2006.

[31] S. van Dongen. A cluster algorithm for graphs. *National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam, Technical Report INS-R0010*, 2000.

[32] D. J. Watts. *Small-worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, Princeton, NJ (USA), 1999.

[33] R. Wille. Restructuring lattice theory: An approach based on hierarchies of concepts. In I. Rival, editor, *Ordered Sets*, pages 445–470. Reidel, Dordrecht-Boston, 1982.