9-2004

Recommender Systems Research: A Connection-centric Survey

Saverio Perugini
University of Dayton, sperugini1@udayton.edu

Marcos André Gonçalves
Virginia Polytechnic Institute and State University

Edward A. Fox
Virginia Polytechnic Institute and State University

Recommender Systems Research:
A Connection-Centric Survey

SAVERIO PERUGINI   sperugin@cs.vt.edu
Department of Computer Science, Virginia Tech, Blacksburg, VA 24061

MARCOS ANDRÉ GONÇALVES   mgoncalv@cs.vt.edu
Department of Computer Science, Virginia Tech, Blacksburg, VA 24061

EDWARD A. FOX   fox@cs.vt.edu
Department of Computer Science, Virginia Tech, Blacksburg, VA 24061

Abstract. Recommender systems attempt to reduce information overload and retain customers by selecting a subset of items from a universal set based on user preferences. While research in recommender systems grew out of information retrieval and filtering, the topic has steadily advanced into a legitimate and challenging research area of its own. Recommender systems have traditionally been studied from a content-based filtering vs. collaborative design perspective. Recommendations, however, are not delivered within a vacuum, but rather cast within an informal community of users and social context. Therefore, ultimately all recommender systems make *connections* among people and thus should be surveyed from such a perspective. This viewpoint is under-emphasized in the recommender systems literature. We therefore take a *connection-oriented* perspective toward recommender systems research. We posit that recommendation has an inherently social element and is ultimately intended to connect people either directly as a result of explicit user modeling or indirectly through the discovery of relationships implicit in extant data. Thus, recommender systems are characterized by how they model users to bring people together: explicitly or implicitly. Finally, user modeling and the connection-centric viewpoint raise broadening and social issues—such as evaluation, targeting, and privacy and trust—which we also briefly address.

> "What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the overabundance of information sources that might consume it."
>
> Herbert A. Simon

1. Introduction

The advent of the WWW and concomitant increase in information available online has caused information overload and ignited research in recommender systems. By selecting a subset of items from a universal set based on user preferences, recommender systems attempt to reduce information overload and retain customers. Examples of systems include top-N lists, book [71] and movie [4] recommenders, advanced search engines [31], and intelligent avatars [5]. The benefits of recommendation are most salient in voluminous and ephemeral domains (e.g., news) and include 'predictive utility' [55], the value of a recommendation as advice given prior to investing time, energy, and in most cases, money in consuming a product. Recommender systems harness techniques which develop a model

of user preferences to predict future ratings of artifacts. The underlying algorithms to realize recommendation range from keyword matching [45] to sophisticated data mining of customer profiles [7]. Recommender systems are now widely believed to be critical to sustaining the Internet economy [105].

Researchers have identified four main dimensions to help in the study of recommender systems: how the system is (i) modeled and designed (i.e., are recommendations content-based or collaborative?), (ii) targeted (to an individual, group, or topic), (iii) built, and (iv) maintained (online vs. offline) [67]. Recommender systems are typically studied along the modeling dimension. The most popular (and over-emphasized) modeling dichotomy is *content-based filtering* [71] vs. *collaborative filtering* [40]. Content-based filtering involves recommending items similar to those the user has liked in the past; e.g., 'Since you liked *The Little Lisper*, you also might be interested in *The Little Schemer*.' Collaborative filtering, on the other hand, involves recommending items that users, whose tastes are similar to the user seeking recommendation, have liked; e.g., 'Linus and Lucy like *Sleepless in Seattle*. Linus likes *You've Got Mail*. Lucy also might like *You've Got Mail*.' Terveen and Hill survey content-based and collaborative filtering systems in a human-computer interaction (HCI) context [109]. Others classify recommender systems from a business-oriented perspective [95], often based on how they are *built*. For instance, Schafer, Konstan, and Riedl survey recommender systems in e-commerce based on interface, technology, and recommendation discovery [95]. These researchers also cast these aspects of recommenders in a two-dimensional space of recommendation lifetime (ephemeral vs. persistent) and level of automation (manual vs. automatic) which is related to how they are *maintained*. Recommender systems, however, have an inherently social element and ultimately bring people together—a viewpoint under-emphasized in the literature—and therefore should be surveyed from this perspective. Accordingly, in this survey, we take a *connection-centric* approach toward studying recommender systems.

To help illustrate the elusive presence of a social connectivity element, consider that the process of recommendation in a 'brick and mortar' setting is inherently dependent on knowledge of personal taste. For example, in a restaurant with a friend, the following dialog might arise: 'The menu looks enticing. Since you are a returning patron, what do you recommend?' 'Well, since you like spicy dishes, you may enjoy the chilli chicken curry.' A mutually reinforcing dynamic ensues. The recommender's personal knowledge of her friend's interests are incorporated into the recommendation process. Conversely, after a recommendation is made, the recipient's personal knowledge of the recommender's reputation helps him evaluate the recommendation. Recommender systems attempt to emulate and automate this naturally social process. This seemingly simple example speaks volumes about the process of making recommendations. Not only does a recommender system have an underlying social element, but its effectiveness is predicated upon its representation of the recipient. Therefore, recommender systems involve user modeling, which includes developing a representation of user preferences and interests. User models can be constructed by explicitly soliciting feedback (e.g., asking the user to rate products or services) [55] or gleaning implicit declarations of interest (e.g., through monitoring usage) [110].

User modeling is directed toward developing a basis to compute overlap, and ultimately is conducted to make *connections* among people to drive recommendation. Thus, once enough users are engaged and modeled to sufficiently sustain a system, connections (recommenda-
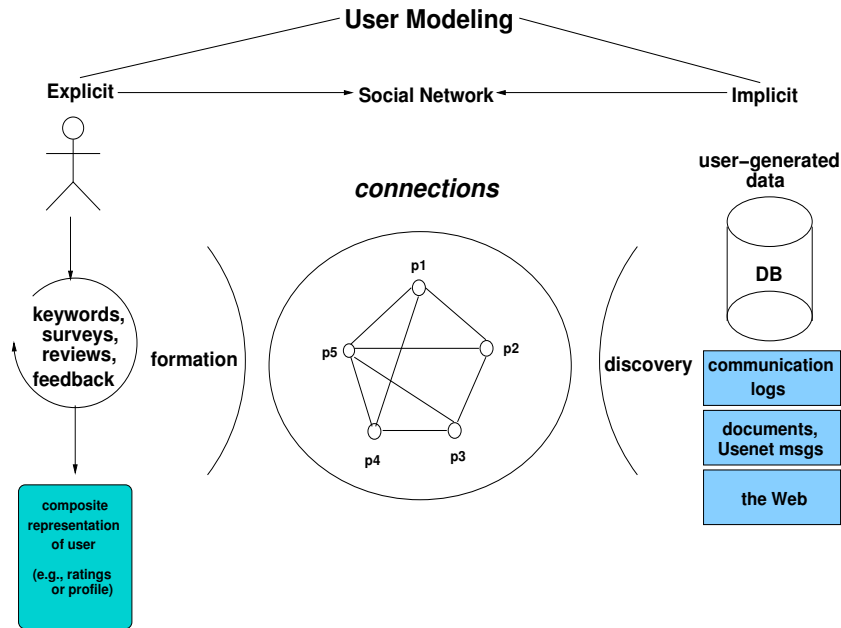
Figure 1. A connection-centric view of recommendation as bringing people together into a social network (center). (left) *Formation* of a social network by *explicitly* collecting ratings or profiles. (right) Identification and *discovery* of a network by exposing self-organizing communities *implicit* in user-generated data such as communication or web logs. Although not illustrated explicitly, these two approaches may be combined.

tions) can be made. Recommendations, thus, are not delivered within a vacuum, but rather cast within an 'informal [community] of collaborators, colleagues, or friends' [57], known as a social network [115]. Explicit user modeling (and correlating the resulting ratings) then can be seen as directed toward forming such connected (community) graph components. Collecting implicit declarations of preference also can be viewed as directed toward inducing social networks. This is analogous to techniques to discover existing social networks from patterns embedded in interaction (transaction) data. Therefore an extension to traditional approaches to implicit user modeling, and an approach toward a basis to compute recommendations, entails directly exposing these self-organizing and self-maintaining social structures. Since social networks model social processes, these informal communities with shared interests are implicit in data generated automatically by electronic communications. This extension is corroborated by a recent trend toward exploring and exploiting connections of social processes in graph representations of self-organizing structures, such as the web, as a viable and increasingly popular way to satisfy information-seeking and recommendation-oriented goals [24, 50, 51]. This less invasive approach not only supersedes the need to explicitly model users individually, but also results in more natural, reflective, and fertile organizations for recommendation. Exploration of identified existing social networks fosters the discovery of serendipitous connections [106], social referrals [57], and cyber-communities [56], and hence offers many opportunities for recommendation. The

use of social networks has expanded to many diverse application domains such as movie recommendation [68], digital libraries [72], and community-based service location [108].

This connection-oriented viewpoint and these two ways of realizing it provide the basis for this survey. We posit that recommendation has an inherently social element and is ultimately concerned with connecting people either directly as a result of explicit user modeling or indirectly through the discovery of relationships implicit in existing data (see Fig. 1). We make connection-based distinctions. Systems are characterized by how they model users to bring people together: explicitly or implicitly. The goal then of a recommender system is to bring as many people together as possible, which also suggests a novel evaluation criterion (e.g., algorithm A connects x individuals while algorithm B connects y) [68]. Thus, while Amazon may make better book recommendations than Barnes and Noble, if they arrive at connected user components in the same manner, then in this survey they would be considered equivalent.

Reader's Guide   The balance of this survey is organized as follows. Section 2 presents an historical perspective of recommender systems and outlines their evolution from IR. Section 3 showcases approaches to creating connections for recommendation via explicit user modeling while Section 4 describes approaches to identifying social networks implicit in (usage) data to explore for recommendation. The relative lengths of these two sections reflect the emphasis each places on connections. Approaches toward identifying implicit communities and resulting systems make social networks salient and thus are treated in greater detail. User modeling and the connection-centric viewpoint raise broadening and social issues, such as evaluation, targeting, and privacy and trust, which we cursorily address in Section 5. We identify various opportunities for future research in Section 6.

2. A Chequered History

While Amazon.com [59], a pioneer in the e-commerce revolution, spear-headed a movement toward recommenders and was instrumental in bringing such systems to critical mass, recommender systems research is a result of a series of shifts in information systems (IS) research. In the 1970s a great deal of IS research was focused on IR. In this era Salton and his students developed the vector-space model [107] and the SMART system [84]. Researchers modeled IR systems with large sparse (and anti-symmetric) term-document matrices which permitted document similarity to be measured by the cosine of the angle between vectors in a multi-dimensional space. Precision and recall became the two quintessential IR metrics [96]. The emphasis of such research and systems was on satisfying short-term information-seeking goals by retrieving information deemed relevant to queries. IR research flourished in this period and many supportive techniques such as relevance feedback [84] were developed, demonstrating qualified success.

As the end of the 1970s drew near, electronic information become more abundant. The 1980s brought a rapid proliferation of information due to desktop computers and applications such as word processors and spreadsheets. In addition, the introduction of e-mail into the mainstream further exasperated the copious amounts of text residing in computers (termed 'electronic junk' by Denning [34]). The new found ease of information gen-

eration ignited a shift in IS research initiatives. Researchers began to focus on removing irrelevant information rather than retrieving relevant information. Information categorization, routing, and filtering became of immediate importance. This first shift spawned an *information filtering* thread.

In 1991 Bellcore hosted a workshop on information filtering (IF) which lead to the December 1992 *Communications of the ACM* special issue on the topic [60]. In this issue Belkin and Croft compared and contrasted IF and IR [14]. While IR entails returning relevant information in response to short-term information-seeking goals via requests such as queries, information filtering involves removing persistent and irrelevant information over a long period of time. Information filtering systems model document features in user profiles [71], which replaced terms in a modeling matrix as a result of this shift (see Table 1). Information filtering later became known as *content-based* filtering to the recommender system community and has been applied to recommend movies [4] and books [71]. Content-based systems model content features of artifacts, rather than of documents, and recommend items by querying such product features against keywords or preferences supplied by the user [49]. SDI (Selective Dissemination of Information), one of the first information filtering systems, was based on keyword matching [45]. Content-based filtering is most effective in text-intensive domains, which account for only a portion of the artifact landscape. Since we take a connection-oriented perspective toward recommendation, content-based models and methods do not find place in this survey.

In addition to identifying these differences, articles in this special issue also reported new research developments. Foltz and Dumais introduced latent semantic indexing as a viable technique to reduce dimensions in a term-document matrix [36]. More importantly for recommender systems, Goldberg et al. coined the phrase *collaborative filtering* [40] while describing Tapestry, which later became known as the first recommender system [86]. Collaborative filtering, which can be defined as harnessing the activities of others in satisfying an information-seeking goal, introduced another shift in IS research. Collaborative filtering entails filtering items for a user that similar users filtered. Instead of computing artifact similarity (content-based filtering), collaborative approaches entail computing user similarity. The most salient difference between these two approaches is that in content-based filtering users do not collaborate to improve the system's model of them, while in collaborative approaches users leverage the collective experience of other users to enrich the system's model. Collaborative filtering is predicated upon persistent user models, such as profiles, which encapsulate preferences and features (e.g., married), rather than ephemeral queries.

This shift replaced features with representations of people (e.g., rating or profiles) to filter documents in a modeling matrix. While documents still constituted the other dimension of the matrix, the word 'document' assumed a broader meaning after the birth of the web. In addition to its traditional interpretation, it also came to mean webpages and bookmarks [25, 85, 110], as well as Usenet and e-mail messages [40, 55].

Collaborative-filtering is effective since people's tastes are typically not orthogonal. However, initially it was not embraced. Meanwhile, the advent of the web and its widespread use, popularity, and acceptance, made reducing information overload a necessity. Of particular importance was social information filtering, a concept developed by Shardanand and Maes [97]. A few years later, in 1996, interest in collaborative filtering led to a workshop on the topic at the University of California, Berkeley. The results of this Berkeley workshop

Table 1. Shifts in matrix models outlining the evolution of recommender systems from information retrieval.

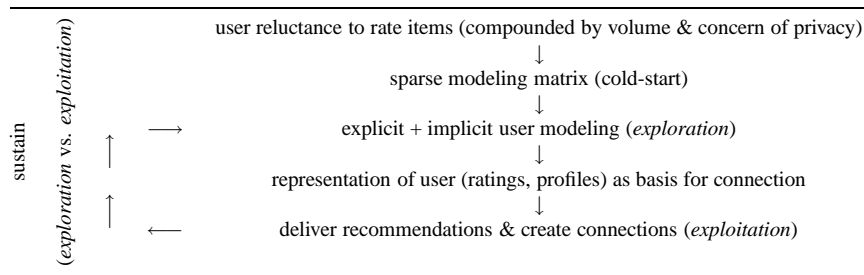| Concept	Modeling Matrix
Information Retrieval	terms $\times$ documents
Information Filtering	features $\times$ documents
Content-based Filtering	features $\times$ artifacts
Collaborative Filtering	people $\times$ documents
Recommender Systems	people $\times$ artifacts

led to the March 1997 *Communications of the ACM* special issue on *recommender systems*, a phrase coined by Resnick and Varian in their article introducing the issue [86].

Resnick and Varian choose the phrase 'recommender systems' rather than 'collaborative filtering' because recommenders need not explicitly collaborate with recommendation recipients, if at all (helping to reconcile the differences between content-based and collaborative approaches) [86]. Furthermore, recommendation refers to suggesting interesting artifacts in addition to solely filtering undesired objects (helping to reconcile the differences between IR and IF). Resnick and Varian define a recommender as a system which accepts user models as input, aggregates them, and returns recommendations to users. Two early collaborative-filtering recommender systems were Firefly and LikeMinds. Firefly evolved from Ringo [97] and HOMR (Helpful Online Music Recommendation Service) and allows a website to make intelligent book, movie, or music recommendations. Firefly's underlying algorithm [91] is now used to power the recommendation engines of sites such as BarnesandNoble.com.

Collaborative approaches constitute the main thrust of current recommender systems research. Once users are modeled, the process of collaborative filtering can be viewed operationally as a function which accepts a representation of users and universal set of artifacts as input and returns a recommended subset of those artifacts as output. More importantly for this survey, recommender systems also are intended to connect groups of individuals with similar interests and to leverage the collective experience rather than merely focusing on the information-seeking goal of a specific individual (as in a typical IR setting). In order to make connections, this function typically computes *similarity* (e.g., closeness, distance, or nearest neighbor). Making recommendations and thus connections then entails approximating this function. Approaches to this approximation that have evolved range from statistical models (e.g., correlating user ratings [55] or reducing dimensions [41]) to attribute-value based learning techniques (e.g., decision trees, neural networks, and Bayesian classifiers) [83] and have demonstrated qualified success [20]. Ultimately these techniques can be viewed as ways to infer structure and induce connections in the modeling matrix space.

This final shift replaced documents with artifacts in the modeling matrix. While the evolution of recommender systems research is characterized by the shifts in matrix models illustrated in Table 1, the sparsity and anti-symmetric properties remained constant across each. As shown below, the web makes the matrix model symmetric. Sparsity is mostly

Table 2. User modeling methodology of a collaborative-filtering recommender system.

sustain	*(exploration vs. exploitation)*	$\longrightarrow$ ↑ ↑ ←——	user reluctance to rate items (compounded by volume & concern of privacy) ↓ sparse modeling matrix (cold-start) ↓ explicit + implicit user modeling (*exploration*) ↓ representation of user (ratings, profiles) as basis for connection ↓ deliver recommendations & create connections (*exploitation*)

attributable to the reluctance of users to rate artifacts. Reluctance results from a lack of time, patience, or willingness to participate. Sometimes the benefits gained from providing constructive feedback are not apparent initially. Reluctance may be partially attributable to a heightened awareness of privacy when divulging personal information. Therefore, collaborative-based recommender systems must mediate an accuracy (of connection) vs. sparsity (of model) tradeoff. The following two sections are devoted to strategies for filling in cells of the initially sparse modeling matrix.

Since 1997 recommender systems research has advanced in many directions, such as reputation systems [87] (e.g., eBay.com), and was placed in a larger context called 'personalization' [79]. The functional-emphasis of current recommender systems makes them 'templates for personalization' [78].

3. Creating Connections: Explicit User Modeling

User modeling entails developing representations of user needs, interests, and taste and is a critical precursor to connecting people via recommendation. In addition to personal characteristics, users can be modeled by their assessments of products in the form of ratings, which then become matrix entries. Sparse user feedback is the single greatest bottleneck of any collaborative-filtering algorithm: 'Collaborative filtering algorithms are not deemed universally acceptable precisely because users are not willing to invest much time or effort in rating the items.' [10]. These problems are compounded in voluminous domains, where a large cumulative number of ratings is required to sufficiently cover an entire set of items. Moreover, as the number of dimensions (e.g., people or products) grows larger, the number of multidimensional comparisons grows. In such situations techniques from data warehousing and OLAP (On-Line Analytical Processing) are applicable [8]. In large domains, users typically examine and evaluate only a small percentage of all items. Shallow analysis of content makes fostering connections difficult since opportunity for user overlap is limited. While in the initial stages of a system, this challenge has been echoed as the 'cold-start' problem [65] (also referred to as the 'day-one' or 'early-rater' problem), it is also ubiquitous during the lifetime of a system. For example, a collaborative recommender has no platform to compute connections for a new user who has yet to rate products or a new item which has yet to be evaluated. Such problems in developing a basis for collaboration provide ample motivation for hybrid approaches which employ content-based filtering in

these specific situations. Hybrid systems have shown improved performance over either single focus (pure) approach [13, 32, 98]. Systems must collect user data which affords the identification of differences, commonalities, and relationships among people. In short, the goal is to add more and more information to transform a sparse matrix to a dense matrix with added structure.

Approaches to user modeling can be studied by how they harvest data [86], either explicitly by asking users to submit feedback through surveys [55] or inferring user interest implicit in (usage) data [28, 110]. Strategies for the former approach are showcased in this section, while those for the later are discussed in Section 4. The most important tradeoff to consider in user modeling is minimizing user effort while maximizing the expressiveness of the representation (as well as privacy). In other words, there should be a small learning curve. Explicit approaches allow the user to retain control over the amount of personal information supplied to the system, but require an investment in time and effort to yield connections. Implicit approaches, on the other hand, minimize effort, collect copious amounts of (sometimes noisy) data, and make the social element to recommender systems salient, but raise ethical issues. The secretive nature of these approaches often make users feel as if they are under a microscope. The user-modeling methodology for a collaborative-based system is illustrated in Table 2.

In explicit user modeling, evaluations [55] and profiles [25] are provided directly by users to declare preferences in response to solicitations for data such as surveys. Evaluations of recommended artifacts can be both quantitative (e.g., ratings), akin to relevance feedback in IR and IF [69], and qualitative (e.g., lengthy reviews at Epinions.com). They also can be positive or negative. In a hand-crafted profile, a user states interests through items such as lists of keywords, pre-defined categories, or descriptions. The system then matches other users against this profile to recommend incoming artifacts. Systems which take such an approach to user modeling are SIFT [119] and Tapestry [40].

Without crossing over to an implicit approach, researchers have identified strategies to deal with reluctance to make an explicit feedback requirement less noticeable and taxing [55, 86]. Possible approaches to motivate users to evaluate items are subscription services, incentives, such as transaction-based compensations, and exclusions [11]. Employing a pay-per-use model for recommender systems, where human experts rate items, is a viable, though less dynamic, option. While this approach connects users through experts and is thus collaborative, it deemphasizes the naturally social (and personal) element to recommenders. Default votes are another way to deal with sparse ratings [20]. Developing and tightly integrating natural user interface (UI) mechanisms to solicit and capture feedback with existing interfaces for recommendation delivery may lead to less intrusive interaction and thus more cooperation and data [39]. A similar approach is to build recommendation into everyday systems, such as e-mail, news, and web clients, and services like collaborative spam detectors (e.g., Cloudmark's SpamNet, http://www.cloudmark.com). In addition to helping to collect more explicit ratings, building recommendation into common UIs may help disseminate recommender systems to the masses. Requiring users to evaluate clusters of, rather than individual, items is another approach to minimizing effort. Rather than tackling sparsity from a user perspective in an explicit approach, it also can be approached from a system viewpoint. Filter-bots which automatically examine and rate all products may occupy empty cells of a modeling matrix [93].

Lastly, a problem endemic to the subjective nature of explicit modeling techniques is that some users are more effusive in their ratings than others. Effusivity in ratings refers to cases of users who share similar preferences, but rate products on completely different scales. Identifying variations in rating patterns is an approach to combat effusivity [10, 37]

Other Considerations   A variety of representations have been used to store user data [22]. The lack of standards to represent such information and its sources (e.g., logs) in a uniform manner make interoperability among recommender systems a challenge [19, 30]. Cookies are mechanisms for capturing and storing user preferences, often employed in e-commerce [18]. While cookies combat the stateless HTTP protocol, like many of these techniques, they raise security and privacy concerns because they are typically unknowingly enabled and as a result personal information is divulged.

A challenge for any user modeling approach (explicit or implicit, for content-based or collaborative recommendation) is the tradeoff between *exploration* (modeling the user) and *exploitation* (using the model to predict future ratings or make recommendations and connections), akin to that in reinforcement learning [88]. Studying the connections which can be made via recommendation and the resulting social network induced in a random graph setting provides technical insight into this problem. Mirza, Ramakrishnan, and Keller identify a 'minimum rating constraint' required to sustain a system and predict values for it based on various experimental rating datasets [68].

Ultimately the approaches to user modeling illustrated in this and the following section are used to connect people. While a purely collaborative approach to recommendation is widely accepted and employed, it is riddled with endemic problems. User modeling must address more than just sparsity. For example, it is difficult to make connections to users with unusual or highly specific tastes. Furthermore, connecting users with similar interests who have rated different items (e.g., 'we both read world politics online, but he ranked BBC.com webpages, while I ranked CNN.com pages') is challenging. Over-specialization of evaluated artifacts, sometimes referred to as the 'banana' problem [26], arises since frequently purchased items, such as bananas in a grocery market basket, will always be recommended. Conversely, some products are seldomly bought more than a few times in a lifetime (e.g., automobiles) and thus suffer from a low number of evaluations. Over-specialization which is grounded in the exploration vs. exploitation dilemma can be addressed by occasionally forcing exploration. For instance, one can inject randomness (e.g., crossover and mutation in a genetic algorithm or epsilon in a reinforcement learning algorithm) into a model. Recommended artifacts also can be partitioned into *hot* and *cold* sets, where the latter is intended to foster exploration and increase the (rating) coverage of items in the system [10].

3.1. Review of Some Representative Projects

The following collaborative-based systems employ many of the explicit user modeling techniques showcased above and illustrate what can be achieved with representations of users. People are connected in the following systems through statistical [41, 55], agent-oriented [25], and graph-theoretic [10] approaches.

GroupLens   GroupLens recommends Usenet news messages [55]. The system models users directly by explicitly eliciting and collecting ratings of messages through an independent newsreader. GroupLens is a project of the recommender systems research group at the University of Minnesota. Usenet news is a personal, voluminous, and ephemeral media (in comparison to movies) and thus an excellent candidate for collaborative filtering. A total of 250 people evaluated over 20,000 news articles [81]. GroupLens takes a statistical approach to making connections. The system predicts how a user seeking recommendation would rate an unrated article by computing a weighted average of the ratings of that message by users whose ratings were correlated with the user seeking recommendation. Correlation is computed with Pearson's r coefficient.

A research issue is deciding whether to provide personalized predictions (as GroupLens currently does) vs. personalized averages. Empirical research using Pearson's r correlation coefficient revealed that 'correlations between ratings and predictions is dramatically higher for personalized predictions than for all-user average ratings' [55]. These results reinforce the hypothesis that not all users are interested in the same articles even within a certain newsgroup (e.g., consider a vegetarian vs. a meat-eater in rec.food.recipes).

Konstan et al. state that 'predictive utility is the difference between potential benefit and risk' [55]. The potential benefit of making predictions is the value of hits and correct rejections. The risk involved in making predictions is the cost of misses and false positives. Konstan et al. identify many of the approaches to increasing rating coverage in explicit user modeling discussed above, such as filter-bots – programs that automatically read and rate all articles [93]. They also identify implicit declarations of quality such as the time spent reading an article. Konstan et al. recognize that some users are more effusive with their ratings than others. The developers of GroupLens began NetPerceptions (http://www.netperceptions.com), a company employing collaborative filtering to provide personalization solutions.

Fab   Balabanović and Shoham take an agent-oriented approach to web document recommendation in Fab, which grew out of a Stanford University digital library project [25]. People are modeled in Fab through explicit (and some implicit) techniques resulting in ratings and profiles. The construction of accurate user profiles drive various agents in dynamically adapting the system. Fab is therefore a representative illustration of the importance and power of user modeling. The hybrid approach to recommendation in Fab retains the advantages of both a content-based and collaborative approach while addressing the disadvantages of each. Moreover the synergy yields new benefits. Fab treats each single focus (pure) approach to recommendation as a special case of its hybrid of the two. 'If the content analysis component returns just a unique identifier rather than extracting any features, then it reduces to pure collaborative recommendation; if there is only a single user, it reduces to pure content-based recommendation' [25].

Fab consists of two processes: collection and selection. During the collection phase, agents gather web sources on topics discovered from clustering user profiles. In the selection process, an agent matches a user profile against what the collection agent has gathered. Thus, one user may be matched against many topics and multiple users may be interested in the same topic. Users proceed to rate results. A user's ratings modifies his profile and

concomitantly helps the collection agent harvest more relevant information for an updated profile. Fab uses

> 'the overlaps between users' interests in more than just collaborative selection. The design of the adapting population of collection agents takes advantage of these overlaps to dynamically converge on topics of interest ... and providing the possibility of significant resource savings when increasing the numbers of users and documents' [25].

Fab connects two users if a collection agent has clustered their profiles in order to collect more web sources of the central theme of the cluster.

Intelligent Recommendation Algorithm   A graph-theoretic collaborative filtering algorithm developed as part of the suite of recommendation engines in the Intelligent Recommendation Algorithm (IRA) program at IBM Research is presented in [10]. The algorithm is motivated by sparsity; Aggarwal *et al.* contend that most collaborative filtering algorithms, such as those for Firefly, LikeMinds, and GroupLens, rely on too many ratings to be successful because they connect users *directly* (e.g., users A and B are connected if their ratings for at least n items are correlated). These algorithms compute *closeness* by taking a weighted average of *only immediate neighbors*. Rather than viewing sparsity as a vice, Aggarwal *et al.* exploit it in their algorithm. The greatest contribution of their algorithm is its use of functional *indirection*, i.e., it allows recommendations to propagate, via more than one intermediary, from a user to another who has not rated common items. The idea is to form and maintain a directed graph, where vertices represent users and edges represent *predictability*. Since the graph is directed, recommendations are anti-symmetric. When one user predicts another, ratings propagate in this model. 'The ultimate idea is that predicted rating of item j for user i can be computed as weighted averages computed via a few reasonably short directed paths joining multiple users' [10]. This effectively makes predictability more general then closeness and addresses the effusivity of ratings.

Aggarwal *et al.* also partition the presentation of resulting recommendations into *hot* and *cold* sets. The hot set, which is two orders of magnitude smaller than the cold, is intended to increase commonality to provide better recommendations while the cold set is intended to foster more exploration and increase the rating coverage of objects. While most of the collaborative filtering systems with explicit user modeling deliver predictable recommendations, this approach encourages *creative* links which violate pre-existing hierarchical classifications to introduce the possibility of *serendipitous* recommendations. Aggarwal *et al.* evaluated their algorithm as well as that for GroupLens, Firefly, and LikeMinds, for accuracy against synthetic data.

4. Discovering Extant Social Networks

Recognition of implicit declarations of user interest is a precursor to discovering existing communities of people. Implicit modeling techniques are a natural extension of those addressed in the previous section. We begin by discussing projects which mine declarations of interest in news [110] and bookmark [85] datasets to cast recommendations and discover

Table 3. Recommender systems research focused on discovering existing social networks. The left column contains modeling concepts, while the center column contains examples of implicit declarations of interest or connections mined from the systems in the right column. Notice that each system relies solely on structural, rather than semantic, information. Note also that the emtpy cell in the lower right hand corner of this matrix is a reflection that few systems take advantage of small-world properties.

Concept	Implicit declaration of interest	System
Traditional Approaches to Implicit User Modeling	URLs in Usenet news bookmarks	PHOAKS Siteseer
Link Analysis and Cyber-Communities	e-mail logs web documents	Discovering Shared Interests Referral Web
Mining and Exploiting Structure	movie ratings datasets hits-buffs, half bow-tie web link topology authorities and hubs bow-tie	Jumping Connections HITS CLEVER
Small-World Networks	actor collaborations author collaborations infectious disease the web	

connections. Although these systems foster communities (as do all recommender systems), they do not make social network identification salient. While active effort is required when extracting declarations of interest to identify connections, natural connectivity among people is self-evident in data. We therefore next discuss mining connections implicit in communication [106] and news [57] logs to induce existing social networks to exploit for recommendation. We then discuss mining and modeling social structure in movie ratings datasets [68] and on the web [52] via link analysis to help identify cyber-communities. We conclude by discussing small-worlds [118], a new class of social networks which present compelling opportunities for serendipitous recommendation. Table 3 outlines the landscape of research showcased in this section.

4.1. *Traditional Approaches to Implicit User Modeling*

Implicit approaches toward modeling users were developed in response to the pervasive reluctance to evaluate recommended artifacts and, although less emphasized, the possibility of building richer representations than with explicit approaches. The idea is to glean user preferences, often secretively, by observation, to serve as surrogates for explicit ratings. A cold-start is less evident in this approach as implicit ratings bootstrap the model and system. Most of the techniques for implicitly gathering and exploiting user information are based on methods and algorithms from machine learning [117, 74, 75] and data mining, which attempt to discover interesting patterns or trends from large and diverse data sources. These techniques are largely based on heuristics. Data mining algorithms also have been
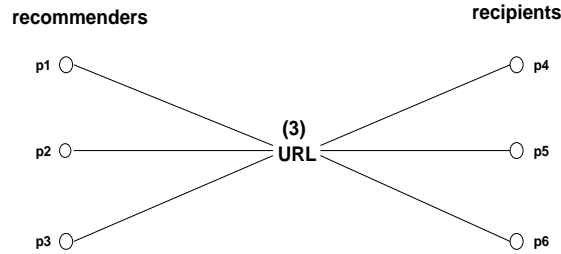
Figure 2. PHOAKS' model of connecting users. The three recommenders (p1, p2, and p3) have mentioned the same URL in Usenet messages and thus that URL is recommended to the recipients (p4, p5, and p6) with a weight of three. This illustration of the recommendation process of PHOAKS makes nine connections among people (each recommender is connected to each recipient and vice versa).

subsequently used to make recommendations. Their expensive time and space complexity is acceptable for user modeling which can be conducted offline. Using data mining techniques for recommendation is a challenge because recommendations must often be delivered in real time [59]. In addition, although not the focus here, inexpensive and less complex techniques for computing recommendations (e.g., correlating user ratings) are relatively effective. Sophisticated approaches to recommendation also suffer from yielding recommendations which are difficult to explain or believe; recommendation explainability and believability are desired properties [46] (see Section 5).

A variety of data sources exist, teeming with and useful for gleaning information about a user's interests and background. Persistent keywords can be extracted from previous user queries. Clickstream data, such as (web) access logs, are invaluable for monitoring, capturing, and chartering a user's interaction with a system, also called 'footprints' [116] (e.g., actions during web browsing, links followed, or amount of time spent on each product page). Web log mining techniques [90] are therefore relevant to this approach and have been used to create a platform to recommend webpages based on browsing similarities with previous users [64]. Web log mining also has been used to trace patterns of navigation to restructure [76] and evaluate [99] websites in a broader personalization context [62]. Other techniques harness UI events such as scrolling and mouse clicking [42]. Alternate implicit indicators of preference include market baskets and purchase transaction data, which are typically exploited by algorithms for mining association rules [2]. Other, less obvious, implicit, self-organizing, and social declarations of interest are bookmarks [85]. The following two projects mine data sources containing implicit declaration of interest for user modeling. Based on their mined implicit ratings they identify connections and make recommendations. Although as a result of implicit user modeling virtual communities are identified, these projects do not make identifying such communities salient.

PHOAKS   Like GroupLens, PHOAKS (People Helping One Another Know Stuff) [110] recommends Usenet news messages, but unlike GroupLens, conducts implicit, rather than explicit, user modeling. PHOAKS interprets the inclusion of uniform resource locators (URLs) in messages as an implicit declaration of interest. The recommendation pro-

cess in PHOAKS entails mining URLs, filtering irrelevant and spurious URLs via heuristics (e.g., remove links embedded into an e-mail signature), and computing a weight for each. A link's weight is its occurrence frequency in messages or, in other words, its number of distinct recommenders. This metric was later extended, formalized, and termed 'authority weight' by Kleinberg [52]. Finally, relevant recommended URLs and associated weights are returned to the user. Precision and recall are applicable to the URLs that PHOAKS recommends and thus were computed. The recommendation process of PHOAKS, which is illustrated in Fig. 2, connects a recommender with a recipient if the recommender has included a URL in a message in the recipient's search topic domain, and if that URL is cited with a high frequency in that domain.

Terveen et al. evaluated their weight metric by comparing PHOAKS recommendations to those provided in frequently asked question lists (FAQs) which are created by human judges of quality. Their evaluation approach suggests a novel application of PHOAKS; it also can be used to semi-automatically create FAQs or recommend improvements to extant FAQs. In addition, Hill and Terveen present an idea for improving the quality of search engine results called community sorted search [48]. The idea is to run a keyword search and cluster the results based on the newsgroups which mention each link in order to disambiguate the query. URLs could be presented sorted by frequency within messages of each cluster.

Siteseer Siteseer models users through their bookmark folders [85]. Bookmarks are a rich data source to exploit for user modeling because they obviate the need to collect explicit ratings and are an implicit declaration of interest, self-maintained, and less noisy in comparison to other implicit rating sources such as a mouse click (which could be random) or a URL embedded in a newsgroup message (e.g., '[URL] is uninformative!') [110]. Furthermore, the binary nature of a bookmark (presence or absence) eliminates the possibility of partial preference. Bookmarks do not capture lack of preference as do other data sources. Most importantly bookmark folders are the basis for the formation of a virtual community. Siteseer computes set intersection between input bookmark folders. A user who has a folder with the greatest overlap with the seeking user's folder is the best qualified recommender for that seeker in the context of that folder. Furthermore each individual URL can be assigned more weight based on the number of folders that it appears in within a virtual community, akin to the authority weight metric of PHOAKS. Siteseer recommends a set of bookmarks in context (i.e., a folder). The system connects people in a directed, non-reciprocal manner akin to IRA [10]. It is pertinent to note that Siteseer does not work on any semantic information such as bookmark title and is therefore an illustration of how indicative purely (social) structural information can be. Siteseer suffers from problems endemic to a purely collaborative approach. For instance, both a new user to the system and an existing user who want to create a new folder provide Siteseer no input basis to compute overlap. Conversely, no collective experience is available to leverage until a cyber-community has been discovered. In addition, bookmarks are typically not public domain or readily available. Siteseer therefore requires trust and buy-in.
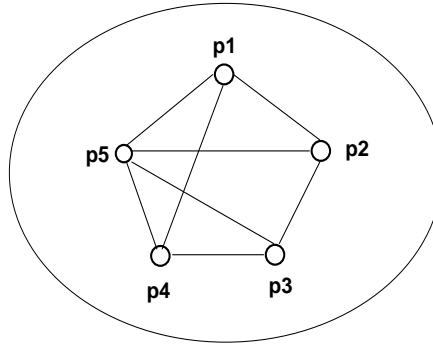
Figure 3. A friendship social network graph. Vertices represent people and an edge between two vertices denotes the 'friend-of' relation.

4.2. Link Analysis and Cyber-Communities

Siteseer is on the cusp of directly identifying communities implicit in data. A natural extension of Siteseer is to proactively mine data which saliently reveals social connections among people. In addition to explicit communities, such as discussion lists, e-groups, and community portals [33], many communities also are implicit in data, such as communications logs [106] and webpages [57], which are fertile reflections of natural connectivity among people. These communities are available to be identified, explored, and exploited [38]. Identification is also worthwhile since, unlike connections induced via explicit modeling approaches and the corresponding systems in the previous section, implicit social networks foster the attractive possibility of *serendipitous* collaborators and recommendation. For example, consider the editor of a journal interested in forming an impartial committee of reviewers for a submitted paper. A social model of author collaborations is an invaluable resource for such a task. Furthermore, unlike other recommender systems which require users to create and maintain profiles [25], approaches which model people connections or social organization implicit in rich, self-generating data result in representations which are likely to be more accurate reflections than a user's perception of his own connections [57]. These communities are typically modeled as social networks [114] and thus research from the social network analysis community is relevant to (automatically) discovering and exploring these networks. In this section we emphasize automatic social network induction, exploration, and exploitation, especially since the manual identification and formation of such communities, collaborators, and referral chains is painstaking, error-prone, and time-consuming.

Social Networks The study of social network analysis dates back to the 1960s. Social networks, which derive their name from social associations among people, model a social process, or specifically, connections among individuals or objects. A social network graph is a unipartite undirected graph, where vertices represent objects (typically people) and edges represent relationships between those objects (e.g., 'friend-of'). A social network

graph is thus characterized by heterogeneous vertices and homogeneous edges, i.e., while each vertex represents a unique entity, each edge represents the same relationship. Fig. 3 illustrates a friendship social network reproduced and enlarged from the center of Fig. 1.

Although not the focus of this paper, it is relevant to note that many interactions and associations in existing web information spaces are modeled via a social network navigation metaphor [57, 113]. Two popular examples are the Internet Movie Database at http://www.imdb.com and the DBLP bibliography website at http://www.informatik.uni-trier.de/∼ley/db/, a collaboration graph of authors, papers, journals, and conferences. In contrast to hierarchical classifications, where users systematically 'drill-down' to hone in on desired information, in sites based on social network navigation users interactively 'jump connections' in support of an exploratory information-seeking goal. Support for foraging in a network, modeling connections among entities, may have been born out of Vannevar Bush's 1945 essay *As We May Think* which foreshadows the web [27]. In this article Bush states that selection by associations among items more closely matches human perception of information foraging than selection by hierarchical index structures.

The increased interest in the concept of social networks has led to the formation of communities and journals devoted to the subject. The International Network for Social Network Analysis (INSNA), which was founded by Wellman in 1978 (http://www.heinz.cmu.edu/project/INSNA), has emerged as an authority on social network analysis. Journals on the topic include *Social Networks*, *Connections*, and the *Journal of Social Structure*. Note however, that the main thrust in the above forums remains targeted toward the sociological, rather than the computing, community. Studying algorithms which identify and exploit the combinatorial structure of social networks is a newly emerging computer science research area being actively investigated by Kleinberg at Cornell University. We refer the interested reader to [51, 113] for a succinct introduction.

Following are examples of projects which mine connections and exploit the resulting social network discovered for various recommendation purposes. The research projects showcased here attempt to identify social networks in data, via various heuristics, implicit in naturally generated data, rather than mining user preferences to form them; they also emphasize modeling connections which distinguishes them from those in the previous subsection. The following two projects emphasis social network induction, exploration, and exploitation.

Discovering Shared Interests   One of the earliest yet untouted attempts at inducing social networks by link analysis entailed analyzing e-mail communication logs, based on heuristics, to uncover an extant social network [106]. In the identified communication network, two individuals share an edge if an e-mail was exchanged between the two. Spurious connections were pruned by heuristics. The discovered social network was intended to help people (vertices) identify others with similar interests and encourage cross-fertilization among the cluster participants. The authors defined the concept of *closeness* between two vertices with the following function:

$$InterestDistance(n_1, n_2) = \frac{|(C(n_1) \cup C(n_2)) - (C(n_1) \cap C(n_2))|}{|(C(n_1) \cup C(n_2))|},$$

where $C(n_i)$ is the set of vertices directly connected to vertex n_i. A value of zero for this interest distance metric indicates that the two input vertices have all neighbors in common. Conversely, a value of one indicates that the two input vertices have no neighbors in common. Such a function is useful for information-seeking activities such as locating a known expert on a particular subject. The authors contend that other individuals within close proximity of that expert vertex are also recommenders on the particular subject. This research was not embraced when published in 1993. However, ten years later, copious amounts of log data as a result of the ubiquitous nature and extensive use of the web has made this work attractive. It is now frequently cited in both the recommender systems and social network analysis literature.

The Hidden Web: Referral Web   Akin to the work described in [106], the Referral Web project at AT&T Labs also implicitly models users to form social networks, where an edge exists between two individuals if their names appear in close proximity in a web document [57]. Again, the underlying assumption is that clustered vertices correspond to people who share similar interests. The Referral Web project also builds on many of the ideas first introduced in [106] by, e.g., exploring the resulting social networks to find experts or recommenders on a particular subject. Kautz *et al.* intended for users to interactively explore the implicit, existing social network in web documents that their mining made explicit. The authors discuss three types of information-seeking goals users could attempt to fulfill in the resulting network—finding referral chains, searching for experts, and proximity search near known experts—which are reflected in the context of computer science researchers in the following information-seeking questions, respectively [57]:

- What is my relationship to Marvin Minsky?

- What colleagues of mine, or colleagues of colleagues of mine, know about simulated annealing?

- List documents on the topic 'annealing' by people close to Scott Kirkpatrick.

Kautz *et al.* also have developed the complementary concepts of *accuracy* and *responsiveness* in a social network [58]. They hypothesized that the accuracy of a referral is inversely proportional to the number of intermediate links between the individual seeking recommendation and the expert providing the recommendation (referred to as 'degrees of separation' – the length of the shortest path connecting two vertices in a social network graph). Kautz *et al.* refer to this in the model they developed as the *referral factor* A, a real number between 0 and 1. $p(A, d) = A^{\alpha d}$, where α is a fixed scaling factor and d is the number of steps from the vertex to an expert. A is the probability that a vertex will refer in the direction of an expert. Likewise, the further removed an expert is from a requester, the less responsive the expert is expected to be. Kautz *et al.* refer to this in the model they developed as the *responsiveness factor* R, again a real number between 0 and 1. $p(R, d) = R^{\beta d}$, where β is a fixed scaling factor representing the probability of an expert responding to a requester d links away. Kautz *et al.* ran a series of simulations on this model with various values for each of these parameters. The results of the experiments reveal a tradeoff between A and R [58]. Through simulation, Kautz *et al.* discovered that

automatic methods can outperform manual referral chaining. Automatic referral chaining however requires sending more messages and thus demands the search of more vertices. Identifying the effects of certain parameters of a developed model for social network graphs is invaluable for setting such parameters when designing a system. Such experimental analysis has been conducted on movie rating datasets [68]. An online demo of Referral Web is available at http://weblab.research.att.com/refweb/working/temp/RefWeb.html.

4.3. *Mining and Exploiting Structure*

Mining social networks from existing data is a method of implicit user modeling for collaborative recommendation. Social networks also can be formed by applying transformations on other, typically bipartite, graph representations identified in datasets. Entertaining the possibility of inducing social network graphs from bipartite graphs fosters social network analysis in domains where the presence of such networks is not salient. Consider that a ratings dataset can be modeled as a bipartite graph rather than a matrix. In social network theory, a bipartite graph is referred to as an *affiliation network* [114] (other researchers refer to them as *collaboration graphs* [44]). In social network theory a *mode* is defined to be a 'distinct set of entities on which structural variables are measured' [114]. A social network graph consists of only one mode (e.g., 'people' in Fig. 3) containing vertices which share a unifying feature, while an affiliation network has two modes—a *primary* mode and a *secondary*—each corresponding to a disjoint vertex set in the bipartite graph.

   A classical example of an affiliation network is the actor-movie collaboration graph (also known as the 'Hollywood graph' [43]), where actor and movie are the primary and secondary modes, respectively. In order to induce a social network graph, members of the primary mode can be brought together via their relationships with members of the secondary mode. The role of the secondary mode is then to bring objects of the primary mode together. This process can be viewed as collaborative filtering. For example, consider that the community of authors of computer science publications is implicit in a computer science corpus or digital library. This social network, where vertices represent authors and two authors share an edge if they have coauthored a paper, can be mined from an author-paper affiliation network representation of the corpus. The Institute for Scientific Information (ISI) maintains such networks and provides associated products and services to help facilitate the discovery process for researchers. An editor of a technical journal may wish to employ a recommender system which models this network to facilitate her formation of an impartial committee of reviewers to evaluate a paper submitted by a particular modeled author. In such cases, candidates would be those with moderate closeness to the reviewee. While an impartial reviewer is one who has never coauthored a paper with the reviewee, a prime candidate also should have moderate knowledge of the submitted paper's topic and thus be slightly close to the reviewee in the resulting social network. In such cases, the editor may want individuals within two or three degrees of separation from the author whose paper is to be reviewed. When applied to such activities, recommender systems which model such social connections among individuals are compelling and applicable to a variety of other important information-oriented tasks (including some studied in the field of 'bibliometrics' [73]) such as granting tenure to a faculty member of a university.
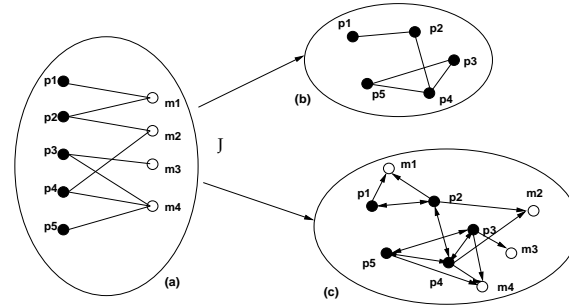
Figure 4. Inducing a social network graph (b) from an affiliation graph representation (a) of a movie rating dataset via a skip jump. Re-attaching movie vertices to the social network graph yields a recommender graph (c) which forms a half bow-tie structure. Figure used from [67] with permission.

Inducing or identifying social networks entails mining social structure in representative datasets which typically results in a graph model. That graph then can be explored and exploited to support many information-seeking and recommendation-oriented activities. Mining structure typically entails identifying characteristics, such as the degree of connectivity or clustering to make statements with certainty about the underlying domain and social process. Investigating why structure arises in the first place also is useful for gathering insight into the underlying social process and its implications on recommendation. Exploring and exploiting graph structures of social processes is a viable and increasingly popular way to satisfy information-seeking goals as reflected in [24, 50, 51]. The following two projects entail explicitly identifying structure, such as connectivity and level of clustering, and in turn exploit it for recommendation. The main idea of this section is to mine, model, and exploit social structure for recommendation.

Jumping Connections   Mirza, Keller, and Ramakrishnan developed a graph-theoretic model to design and evaluate recommender systems [68]. Their approach is connection-centric and entails inducing social networks and identifying various structural properties therein from public domain movie rating datasets. Each ratings dataset $\mathcal{R}$ used, namely EachMovie (collected by the Digital Equipment Corporation) and MovieLens (developed by the recommender systems researcher group at University of Minnesota for the Movie-Lens project which is based on the ideas from GroupLens; see http://movielens.umn.edu), was a matrix (people $\times$ movies) of ratings and was modeled as an undirected bipartite graph, where the two disjoint vertex sets correspond to people (P) and the movies (M) they have rated (Fig. 4a). An edge between a vertex of each set denotes the 'rated' relationship. In these affiliation networks, people and movies are the primary and secondary modes, respectively.

Mirza *et al.* induced various social networks by applying various 'jump' specifications of how individuals in the affiliation network representation of $\mathcal{R}$ are to be connected in the resulting social network graph G_s (Fig. 4b). A jump is a function $\mathcal{J} : \mathcal{R} \mapsto S$, where $S \subseteq P \times P$. The elements of S (unordered people pairs) represent the edges of G_s.

There are many ways of jumping. A *skip jump* directly connects person x and person y in G_s if they have rated a common movie. The skip jump is a special case of a *hammock jump*, which induces a G_s where two people are directly connected if they have evaluated w movies in common, where w is termed the *hammock width*. A jump can be thought of as a recommendation algorithm and therefore, all recommender systems bring people together, in one form or another, via jumps, albeit each system may jump in strikingly different ways. Systems then can be classified by how they jump and the number of people each jump connects in the resulting G_s. Mirza *et al.* also propose the number of people connected by a jump as an evaluation metric for recommender systems (see Section 5). Finally, re-attaching the movie vertices to the people vertices of G_s (Fig. 4b) produces a recommender graph G_r (Fig. 4c), where directed paths from people to movies may be discovered.

After applying a jump, Mirza *et al.* attempt to identify structural properties in G_s and G_r. They have identified a half bow-tie structure in G_r. This higher-order structural observation is analogous to the identification of the (full) bow-tie structure in the link topology of the web [21], which arises since the web's nucleus consists of a strongly connected component (SCC, the center of the bow-tie), webpages which link only into it (the left side of the bow-tie), and pages which are only linked to from the SCC (the right side of the bow-tie). In G_r, the G_s is the SCC (the center of the half bow-tie) while the movie vertices of G_r, the right half of the bow-tie, are only linked from this SCC. Fig. 4 illustrates the entire process by which a G_s is induced from an affiliation network representing $\mathcal{R}$ via a skip jump and subsequently augmented to G_r by reattached vertices of the secondary mode of $\mathcal{R}$ (movies) to G_s. Note that people are brought together via movies analogous to Kleinberg's authorities being brought together via hubs in the affiliation topology of the web [52].

While both data sets studied by Mirza *et al.* are extremely sparse ($\geq 93\%$), both are connected. Another interesting discovery was that each dataset exhibited a *hits-buff* structure, i.e., some people (buffs) rate all movies while some movies (hits) are rated by all people. This intriguing structural property is attributable to both the underlying domain and the hypothesized power-law distribution [29] followed by movie ratings (and other self-organizing systems, e.g., the web and airports). Notice again that this structural discovery resembles the mutually reinforcing relationship of Kleinberg's authorities and hubs.

The power law distribution of movie ratings has other implications. For example, as w increases, G_s becomes more disconnected. A hammock width of seventeen or greater disconnects G_s in the MovieLens dataset. What is surprisingly interesting however is that at that breaking point the graph has one large SCC plus many isolated vertices rather than many small SCCs. Mirza *et al.* refer to this process as 'shattering' the graph. This phenomenon is again attributable to the power-law distribution in movie ratings. One large SCC and many small SCCs (i.e., communities) do not emerge when G_s breaks because movie viewers typically do not have strong biases in their tastes. Mirza *et al.* contend that in other domains, such as books and music, small communities representing the many diverse genres (e.g., country or jazz) in such domains will arise as w is increased. Thus, the group hypothesizes that books and music ratings do not follow a power-law distribution. Such hypotheses have yet to be verified due to the inability to obtain datasets in such domains, and thus leave scope for future work.

Akin to the tradeoff between the referral and responsiveness factor in Referral Web [58], Mirza *et al.* report the effect of the hammock width w on the minimum rating constraint κ (i.e., the minimum number of items users must rate prior to receiving recommendations) and the various shortest paths used to route recommendations, such as l_{pp} (shortest path length in G_s), l_r (shortest path length in G_r), and l_{pm} (shortest path length from people to movies in G_r). Identifying an optimal number of evaluations required to sustain a system is invaluable from a design perspective. Random graph theory plays a large role in this analysis since movie ratings do not correspond to a particular graph, but rather a family of graphs. Thus, while graph algorithms, such as Dijkstra's single source and Floyd's all pairs shortest path algorithms, can be applied, they will not reveal any interesting properties in $\mathcal{R}$. The researchers thus study random graph models for recommender dataset graphs and associated attributes such as degree distributions. Random graph models typically accept an edge probability and number of vertices as input and output a random graph meeting such properties and constraints.

The ultimate goal of the Jumping Connections project is to develop a model wherein the implications of certain parameters (e.g., w or κ) on the structural properties of G_s and G_r guide the design of a recommender system. Development of such a model entails studying the effects certain jumps have on the properties of the resulting G_s. Based on random graph models the researchers would like to say that G_s is connected if some condition holds (e.g., $\kappa > 20$). Being able to make such statements with high probability is sufficient from a theoretical computer science point of view. Such information is critical not only to comparing various recommender systems but also to providing designers answers to questions such as 'If I know that I only need a hammock width of four to effectively make recommendations, to what should I set the minimum rating constraint of our new recommender?'

Future work includes incorporating user ratings. Thus far the group has only been considering the binary nature of ratings (presence or absence) to make connections. This again reflects that substantial insight into recommendation can be achieved with purely structural information, as in Referral Web [57] and Siteseer [85], rather than semantic information such as ratings. Mirza et al. also would like to formally test their hypothesis that jumps (i.e., recommendation algorithms) never bring disconnected components of the recommender dataset graph together. The researchers hypothesize that in certain pathological cases, such as when the dataset graph contains two isomorphic subgraphs, the singular value decomposition [104] (also known as latent semantic indexing to IR researchers [15]) will connect disconnected portions of $\mathcal{R}$. Jumping Connections is a novel research project; it helps make a science out of recommender systems which have traditionally lacked any sophisticated model to design and evaluate systems.

HITS   In addition to mining bookmark, communication, and news datasets, recently much research has been conducted on mining the link topology of the web [31]. The most significant and compelling contribution in this area is Kleinberg's observation, through mining link structure, that the web is an affiliation network consisting of an *authority* mode and a *hub* mode. Authorities are authoritative sources on a topic (e.g., PGA.com for golf) while hubs are collections of links to authorities (e.g., bookmarks or favorite links pages).

Hubs and authorities mutually reinforce each other: good authorities are linked to by many good hubs, while good hubs link to many good authorities [52].

The identification of this compelling structural property was important since the web was widely believed to be a unipartite graph where all pages were perceived to be of the same type. This observation gave birth to search engines which rely exclusively on purely structural information to navigate and search the web. The identification of a second mode of webpages, namely hubs, was required in order to connect authorities who would otherwise not be linked due to competitive reasons (e.g., the webpage of Microsoft does not link to the webpage of IBM, yet both are computer companies). Kleinberg's HITS (Hyperlink-Induced Topic Search) algorithm exploits such structural information to search the web for authoritative sources.

The HITS algorithm attempts to identify good hubs and authorities by assigning hub and authority weights to webpages based on a (QR [104]) matrix power iteration. A similar project [56] approaches trawling the web for cyber-communities as mining structure in bipartite graphs. These researchers however describe Kleinberg's hubs and authorities as fans and centers, respectively. The two groups have collaborated on a survey of the measurements, models, and methods of the 'web graph' [50]. Amento, Terveen, and Hill experimentally measured whether authoritative sources are good predictors of quality [9].

As a result of mining the link topology of the web, the HITS algorithm is the most illustrative and powerful example of what can be done purely with structural information akin to Referral Web [57], Siteseer [85], and Jumping Connections [68], rather than semantic information, such as text indexed on a page. Specifically, HITS is evidence that link structure is sufficient to correctly characterize, aggregate, and leverage the interests of a large population.

The CLEVER search engine of IBM [31] was designed based on HITS. Google, developed at Stanford University, is another search engine which considers link structure [23]. PageRank, the algorithm of Google, only computes authority weights, however, and thus does not connect authorities via hubs. The Google engine also analyzes textual information in addition to link structure. It therefore involves a hybrid approach (i.e., both structural and semantic information is incorporated). The most salient difference between the PageRank and HITS algorithms is that PageRank analyzes the link structure of the web offline while CLEVER mines the web on a per query basis. An implication of this difference is that Google is a much more practical application than CLEVER due to the expensive matrix operations required for HITS in real-time.

4.4. Small-World Networks

While typical examples of social networks are the actor collaboration graph and author collaboration graphs, a new class of social networks (and associated random graph models) has emerged – *small-world networks* [44]. These networks naturally model the small-world phenomenon. In the late 1960's, Harvard social psychologist Stanley Milgram paved the way for small-world network analysis by conducting a unique chain letter experiment [66]. As opposed to the other projects discussed in this section, rather than attempting to discover a social network, Milgram hypothesized that a social network existed and tested both if that
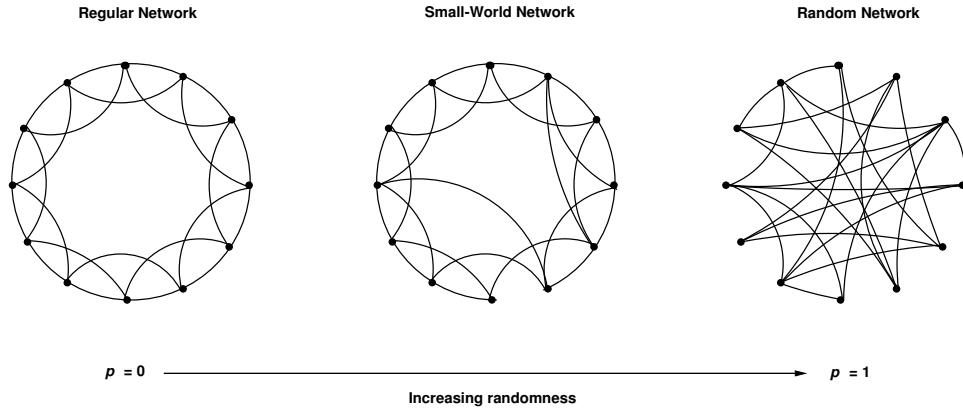
Figure 5. The Watts-Strogatz model for small-world graphs. Edges from a regular ring lattice (left) are randomly rewired based on a rewiring probability p, yielding a small-world network after only a few random rewirings (center). When p=1, all edges are rewired yielding a completely random graph (right). Figure used from [67] with permission.

network exhibited small-world properties and, more importantly, if individuals, with only local views of the network, could successfully construct short chains between members.

The experiment involved source individuals in Nebraska or Kansas delivering a letter to a target person in Boston, MA via intermediaries. Source individuals were given only a few cursory biographical characteristics of the target and permitted to forward the letter only through individuals to which the source was on a first name basis and so on. As letters propagated east across the US, the experiment revealed that any two individuals in the acquaintance network of the United States could be connected through a few intermediaries; or more formally that the acquaintanceship network of the US exhibited small-world properties. Milgram's experiments specifically revealed that any two randomly picked individuals residing in the US were connected by no more than six intermediate acquaintances. The small-world phenomenon later became popularly known as 'six degrees of separation,' after which both a play and its movie adaptation have been named. The degrees of separation between two vertices in a social network graph is the length of the shortest path connecting the two vertices. The problem of reducing the number of intermediaries in a social network has historically been referred to as *information routing*.

When put into context the small-world phenomenon does not seem outlandish. Consider that a college student in a large state university might be connected to the president of the United States through five intermediaries on a first name basis (e.g., student – professor – department chair – dean – university president – governor – president of the US) in a connection path of length six. The impact of the small-world phenomenon is ominous when studied from the perspective of infectious disease.

A small-world is a graph G which exhibits certain structural properties while modeling some natural phenomenon. Small-world graphs are structurally characterized by sparse edges (i.e., many more vertices than edges), highly clustered vertices, and relatively short paths between any two vertices. Random graph models have been proposed to study small-

world networks [3, 6, 12, 35]. Watts and Strogatz have developed a random graph model for small-world networks based on a rewiring probability (see Fig. 5) [118]. The Watts-Strogatz model for small-world graphs interprets a small-world network as a hybrid between a completely ordered ring lattice (wreath) and a completely random graph, leaning closer to the lattice. The *average minimum path length* A of G is 'the minimum path length L averaged over all pairs of vertices' [44]. Watts and Strogatz refer to average minimum path length as 'characteristic path length' [118]. The *clustering coefficient* c of a vertex in G is the degree of the vertex divided by the maximal number of edges which could possibly exist between the vertex and all its neighbors. Characteristic path length is a global property of a network, while clustering coefficient is a local property measuring the 'cliquishness' in a neighborhood. Watts and Strogatz discovered that by replacing some local contacts with arbitrary ones (called 'random rewiring'), the clustering coefficient of a network remains high, close to that of the initial completely ordered network, while the characteristic path length is drastically reduced. The model begins with a regular ring lattice where vertices are highly clustered with large average minimum path lengths between any two vertices (Fig. 5, left). In the model, from this lattice edges are randomly rewired based on a rewiring probability p. Randomly rewiring only a few edges (less than ten percent of total edges) renders a small-world graph where vertex clustering remains relatively high, but average minimum path length is drastically reduced in comparison to the ring lattice (Fig. 5, center). When all edges are rewired the model produces a completely random graph where vertex clustering is low and average path length is small (Fig. 5, right). The replacement of local links with random links (random rewiring) manifests itself in the real world when someone moves to a new city, starts a new job, or joins a club.

Watts and Strogatz hypothesized that small-world properties exist in diverse domains and that the small-world phenomenon arises in many self-organizing systems such as the actor-collaboration graph, the power grid of the Western US, and the nervous system of the nematode worm Caenorhabditis elgans. They examined these real world networks to test the existence of small-world properties. All three observed networks exhibited small-world properties. These conclusions suggest that the small-world phenomenon is common in many large networks found in nature and not merely an artifact of an idealized world. Small-worlds also are believed to exist in the spread of disease, and most importantly as of recently, the web [44, 118].

The research of Watts and Strogatz has peaked interest in studying many other diverse social domains via small-world graphs. By conducting an exhaustive survey of the actor-collaboration graph implicit in the Internet Movie Database, Tjaden at the University of Virginia found the maximum degrees of separation from any actor in Hollywood to American film actor Kevin Bacon, termed the 'Bacon Number,' to be seven (four on average, and eight from all film actors and actresses of any nationality) [112, 44]. Tjadan maintains a website called the *The Oracle of Bacon* at http://www.cs.virginia.edu/oracle/ which provides an interface to compute Bacon numbers.

Determining the impact the small-world phenomenon has on the dynamic behavior of a distributed system is an open research issue. If small changes to an edge set of a graph can have a dramatic impact on its global structural properties, the same changes might affect its behavior as well. This issue was studied in the context of infectious diseases and games. With infectious diseases, shortest path length implies faster spreading of the

disease. Therefore, global dynamics do depend on coupling topology. Games on graphs, such as the Prisoner's Dilemma reward game, can be similarly analyzed. When a low level of 'hardness' in players exists, cooperation dominates, but the timescale depends quite sensitively on the fraction of shortcuts. When the level of hardness in players is average, the amount of cooperation which succeeds depends on the fraction of shortcuts. Cooperation is more difficult to sustain when the number of shortcuts increases due to those individuals who do not desire to cooperate. One would like to optimize both the spread and sustenance of cooperation.

Another important open research question is whether individuals with only local knowledge can effectively navigate in a small-world [53]. In addition to identifying small-world properties in the acquaintance network of the United States, Milgram's experiment, more importantly, illustrated that people with only local knowledge of the network (i.e., of their immediate acquaintances) were actually successful at manually constructing acquaintance chains of short length. An open research question is if computers, as opposed to humans, can construct short referral chains via an algorithmic procedure. Kleinberg developed a decentralized algorithm which is capable of constructing paths of short length in a small-world for one particular model of small-world networks which he developed [54]. Electronic communications (e.g., e-mail) have made Milgram's experiment and research results easily replicable. For example, the Small World project (http://smallworld.columbia.edu), which is led by a group of researchers including Watts, is an online experiment to test Milgram's six degrees of separation hypothesis.

Adamic tested the web and discovered through experimentation that it is a small-world, where vertices correspond to websites, as opposed to individual webpages, and hyperlinks are considered to be undirected [1]. She also addressed the implications these properties have on searching the web and on discovering the structure of certain social communities with a web presence. L was estimated by averaging the paths in breath-first search (BFS) trees. L was small (about 3.1 hops) and C was 0.1078 compared to 2.3×10^{-4} for a corresponding random network with the same number of vertices and edges. A second case study considered directed links. The largest SCC was computed on the graph. BFS trees were formed on a fraction of the vertices to sample the distribution of distances. L was slightly higher due to the directed links and C was 0.081 compared to 1.05×10^{-3} for a corresponding random graph with the same number of vertices and edges. This empirical evidence reveals that there is a small-world network of websites. A third case study was performed on the .edu subgraph of the web which is considerably smaller (i.e., distances between every vertex could be actually computed). Again, the largest SCC was computed. L was found to be 4.062 and C was found to be 0.156 vs. 0.0012 for a corresponding random graph with the same number of vertices and edges. Hubs may foster short average path lengths between two randomly selected (authority) webpages and thus may be integral to small-world properties in the web. 'In summary, the largest SCCs of both sites in general and the subset of the .edu sites are small-world networks with small average minimum distances' [1].

Adamic discusses how search engines could take advantage of these small-world properties. The idea of capitalizing on these structural features of the web for web search is that it is more advantageous to return good starting points, called 'centers,' 'index pages,' or 'hubs,' i.e., where 'the distance from them to any other document within the group is

on average a minimum' [1]. Search results thus can be grouped according to such good starting points. An application built around these ideas presents a user with a list of centers sorted by the SCC size and allows the user complete freedom in exploring these SCCs.

The next important research question is, aside from identification and navigation: Can a small-world topology reveal clues to the structure of real world communities, via the structure of the connections between documents which members of those communities manually created? Adamic contends that a small-world topology can provide clues to the structure of real-world communities. She states that 'exploring the link structure of documents which belong to a particular topic could reveal the underlying relationship between people and organizations' [1]. Adamic discovered, via the application she developed, that the pro-life community in the real world is larger, more closely knit, and better organized than the pro-choice community by calculating the number of sites contained in each SCC. Such an observation can have significant implications for marketing strategies and red ribbon campaigns.

In summary, small-worlds present opportunities for recommender systems. If identified, not only do they help model users and communities implicitly by revealing social structure, but also help connect people via short chains. For example, if search engines could take advantage of the web's small-world property, then users with only local knowledge of the web may actually be able to find and construct short paths between pairs of webpages. Since Albert, Hawoong, and Barabási have shown the diameter of the web to be nineteen [3], if one knows where one is going, then one can get there fast. The *diameter d* of a graph G is the 'shortest path between the most distant vertices' [44]. Currently web search engines do not exploit short path lengths between webpages. Finding such short paths within information abundant spaces, akin to using a compass, reduces information overload and expedites recommendation. In conclusion, the main point to take from this section is that knowledge of the existence of certain, typically social, structures and properties (e.g., connectivity, bow-tie, or small-worlds) can be exploited by recommenders, such as search engines, to intelligently provide more effective results.

5. Broadening Issues

As we have illustrated, recommender systems are not used in isolation but are rather cast in a broad social context. In this section we briefly discuss broadening issues regarding recommender systems, such as evaluation, targeting, and privacy and trust. This coverage of broadening aspects of recommendation is not meant to be exhaustive. While each issue warrants survey in isolation, we only make some cursory remarks here. This section is intended to provide pointers to authoritative sources to the reader interested in how recommendation affects these topics.

5.1. Evaluation

While evaluation is critical to the success of a recommender system or algorithm, rigorous evaluations are rarely performed, mainly due to the lack of universally accepted formal methods for system evaluation. Analyzing a recommendation algorithm from a functional perspective is the most popular approach to evaluation [20]. Such an approach typically

involves measuring 'predictive accuracy' via a training/test set analysis [41]. An alternate, more personal and social, approach is to conduct HCI studies with user participants [101]. The former approach is employed more frequently than the latter. These two approaches are at opposite ends of an evaluation spectrum and are referred to as off-line vs. on-line evaluation [47].

Functional-oriented Evaluations   Most evaluations of recommender algorithms use standard IR metrics like precision and recall [110]. The GroupLens recommender systems research group has evaluated recommendation algorithms for e-commerce [94]. They investigated traditional data mining techniques (e.g., association rule mining in transactional data [2]), nearest-neighbor collaborative filtering, and dimensionality reduction. Again, the evaluation metrics discussed are traditional: support and confidence [2], and precision and recall [96].

After [86], the second most highly cited article in the recommender systems literature is a paper describing empirical evaluation of the predictive accuracy of collaborative filtering algorithms [20]. Breese, Heckerman, and Kadie focus on predictive accuracy rather than efficiency. Their approach to evaluation is purely statistical. They developed two classes of evaluation metrics to characterize accuracy: average absolute deviation of predictions and utility of ranked lists of recommended items. Breese *et al.* compute these metrics across popular functional approaches to collaborative filtering, such as correlation, Pearson's r, vector similarity, inverse user frequency, and statistical Bayesian methods, in various ratings datasets, including the EachMovie dataset. Although human satisfaction is not considered in this analysis, the evaluation techniques introduced in [20] have become a de facto standard for recommender systems [41].

While there is nothing intrinsically bad about these functional-oriented approaches to evaluation, they do not capture the underlying social process involved in recommendation. As echoed repeatedly in this survey, recommendation is an inherently social process and recommender systems ultimately connect people. Evaluating a recommender system via a purely mathematical analysis of its functional approximation (predictive accuracy) ignores this integral social process [47]. In addition, recommendations must be explainable [46] and believable to users. Functions are not always explainable; a function could be a black box to a user, which provides no transparency [46]. Therefore, traditional training/test set approaches [41] and associated metrics, such as precision and recall, and support and confidence, are inadequate for algorithm evaluation from a social perspective.

Social-oriented Evaluations   Swearingen and Sinha posit that the effectiveness of a recommender system transcends the predictive accuracy of its underlying algorithm [101]. The opposite end of the recommendation evaluation spectrum is purely social and entails conducting satisfaction surveys and studies. Such studies are rare largely because of the high cost involved. Sinha and Swearingen conducted studies with users, as part of the HUBRIS (HUman Benchmarking of Recommender Systems) project at UC Berkeley, directed toward comparing recommendations given by friends to those by six commercial and widely available recommender systems [100]. Their study also addressed the degree to which assessments of overall recommender system performance were correlated with the

quality of the recommendations or the user interface. The ultimate objective of their study is to develop a user-centered design approach toward recommender systems. Their initial results indicate that an effective recommender inspires user trust in the system, provides explainable recommendations w.r.t. system logic (termed 'transparency') [102], and makes serendipitous connections.

A Hybrid Approach   In part to reconcile these two extreme approaches toward evaluation, Mirza, Keller, and Ramakrishnan developed the Jumping Connections model to evaluate recommendation algorithms based on the number of people they connect. For example, recommendation algorithm A can be compared to recommendation algorithm B by the number of people connected by the jump that A models vs. the number of people connected by the jump that B models. Such an approach places an emphasis on the underlying social element to recommendation and the social network induced as a result.

The most striking contribution of their approach to evaluation is that it resides at the fringe of HCI. Using 'the number of people brought together' by a particular jump as an evaluation metric for recommender systems is not as complex or sophisticated as studying evaluation from a traditional functional-oriented perspective, but is more cost effective than an empirical study. Moreover, it is not as social-oriented as a full blown study with user participants.

5.2. *Targeting*

Targeting answers the question 'for who or what are we building this system?' and can be critical to the success of a recommender system. Systems can be targeted to, e.g., all users, a particular user, a set of topics, or per user per topic. A lattice induced by these choices of targeting dimensions is shown in Fig. 6. We briefly mention an example of each strategy in the targeting lattice. Other possible targeting dimensions include geographic location or genre.

Top-N lists and FAQs are designed to provide recommendations targeted to all users (Fig. 6, bottom; the greatest lower bound). 'My Sites,' such as MyYahoo! [70] are customizable web portals targeted to the individual (Fig. 6, left). The user selects areas of interest from several content modules, such as news, stock quotes, weather, and sports, and the system in turn provides recommended links to these topics. IndexFinder [76] adapts websites based on various niche topics (Fig. 6, right). The system automatically synthesizes index webpages (i.e., hubs) consisting of links to other webpages covering a particular common topic. The synthesis of index pages is conducted by mining user browsing patterns implicit in web logs and conceptually clustering webpages to induce topics. The 'Syskill & Webert' project [77] combines the individual and topic targeting dimensions by targeting per user per topic (Fig. 6, top; the least upper bound). A user rates visited webpages on a three point scale and the system induces a user profile by analyzing the information on each webpage. The system then recommends interesting websites to users based on the learned profile and webpage topic.
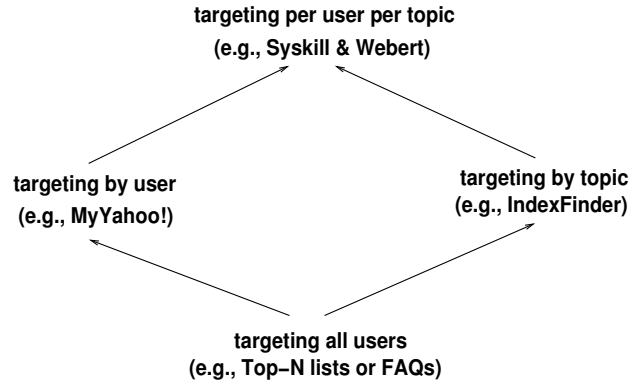
targeting per user per topic
(e.g., Syskill & Webert)

targeting by user   **targeting by topic**
(e.g., MyYahoo!)   **(e.g., IndexFinder)**

targeting all users
(e.g., Top–N lists or FAQs)

Figure 6. Targeting a recommender system. A lattice structure is induced by choices of targeting dimension: to all users, a particular user, a topic, or per user per topic.

5.3. Privacy and Trust

As expressed throughout this paper, recommendation raises issues of privacy and trust [80]. Both an explicit and implicit approach to user modeling suffer from privacy issues with the latter assuming more responsibility for evoking concern due to its elusive nature.

A tradeoff exists between collecting and leveraging as much user information as possible and inspiring trust between the user and system. Storing users' profiles and background leveraged for recommendation on the client-side is an approach to empowering users with more control over their personal information and degree of privacy. Users then control the tradeoff between benefit and risk by deciding on their desired level of involvement. This sentiment is corroborated by Singh, Yu, and Venkatraman [108] who compare community-based networks and recommender systems to suggest that people really want to control who sees their ratings and understand the process of recommendation. Belkin however argues that with sufficient reason to trust recommendations users are willing to give up some measure of control and accept suggestions [16]. Some researchers even advocate exposing user profiles to build trust [120]. Other researchers contend that understanding why a recommendation was made cultivates trust in a system [102]. Trust also can be designed into online experiences [92].

The issue of user privacy in general is not specific to a purely collaborative approach. Privacy and trust issues however are compounded in a collaborative setting as user interests, background, or identity may be exposed to other users participating in a social network in addition to the system. Moreover, issues of privacy are not confined to a user and a recommender system, but rather cascade across the social networks such systems induce or identify. Recall that when a recommender ratings dataset shatters, many isolated communities take form. As the hammock width increases initially, clusters tend to be fuzzy with many vertices bridging across each cluster. As the hammock width continues to increase, however, clusters will crystallize with only one or two vertices serving as bridges across SCCs. Often one or two users span such communities and provide the basis for a serendip-

itous recommendation. Such users or bridge vertices are called 'weak ties' in the social network analysis community [114]. Small-worlds are teeming with weak ties which are critical to the small minimum average path length of the network. Weak ties help connect two or more strongly connected components. While weak ties are important to providing serendipitous recommendations, just knowing that one exists (possibly by receiving an un-expected recommendation) compromises the privacy of the person (the weak tie) which fostered the connection [82]. For example, consider an avid music fan who only rates CDs of Italian operas. A recommendation of Indian classical music for such a user implies that there exists at least one individual spanning these two communities. The rare nature of the tie compromises identity.

6. Future Work and Conclusion

The field of recommender systems is young and evolving. We have identified some directions for future work.

- **Distributed Recommendation Infrastructure and Interoperability**: Taking recommendation out of specific systems and casting it in a broader, distributed information infrastructure is a direction for future research [61]. Such an infrastructure [30] fosters the possibility of users managing and maintaining their own client-side user model and context to share at their discretion with participating recommenders. Addressing and resolving technical issues, such as interoperability and standardization, and social issues, such as buy-in, is essential to the realization of this vision.

- **Formal Recommendation Modeling and Design Methodologies**: As with any young, evolving, and multidisciplinary field, recommendation is lacking unified methodologies to study, design, and evaluate systems. Without such methodologies, unsystematic development will continue to persist resulting in kludge with consequential problems of cost and interoperability. Initial steps toward such models are reflected in projects such as Jumping Connections [68] and the on-line evaluation framework [47]. Design patterns and languages may help capture solutions to recurrent modeling problems, such as sparsity and privacy, and foster the semi-automatic construction of systems.

- **An HCI Approach: Designing for Interaction**: Developing and integrating less in-trusive and salient interfaces for explicit product evaluations and ratings with recommendation delivery interfaces is a compelling line of future research. The interest and need for such research, including possibly usability evaluations, on extant and new streamlined UIs for recommenders systems, is reflected in an *ACM Transactions of Computer-Human Interaction* call for papers [111]. Furthermore, studying user inter-actions with recommender systems is becoming an increasingly popular way to design such systems [103]. HCI researchers involved in this effort are optimistic that results of such studies will lead to general design guidelines. These two directions present op-portunities for the HCI community to make inroads to recommender systems research which needs such expertise.

- **Information Appliances**: Computing is becoming increasingly ubiquitous. Physical computing devices and 'information appliances,' computer-enhanced devices dedicated

to a restricted suite of information-oriented tasks, are no longer a vision, but rather a reality [17]. The scope of information appliances is no longer limited to handheld computers, PDAs, and mobile phones, but has extended to MP3 players and watches. In addition, ubiquity is enriched by voice applications and portals (e.g., Tellme, http:// www.tellme.com), collectively called the 'voice web' [89], avatars [5], and information kiosks [63]. These devices present compelling opportunities to deliver 'recommendations on demand.' For example, consider delivering a restaurant recommendation and associated coupon on a cell phone to a businessman in search of a vegetarian meal while waiting in an airport. Leveraging the ubiquity of such notification systems for user modeling and recommendation delivery is an open research issue.

In conclusion, we have provided a survey of recommender systems research according to the way they model users and resulting connections they achieve or identify. We feel this is a more holistic approach to recommendation as it captures the underlying social element in all recommenders. The reader will have noticed that intelligent techniques to model users are slowly being augmented with approaches to exploit the combinatorial social structure implicit in usage data. The future of recommender systems will ultimately lie in mediating these two approaches and developing unified methodologies to systematize the process of representing users and building systems.

Acknowledgments

References

1. L. A. Adamic. The Small World Web. In S. Abiteboul and A-M. Vercoustre, editors, *Proceedings of the European Conference on Digital Libraries (ECDL'99)*, Lecture Notes in Computer Science 1696, pp. 443–452, Paris, France, September 1999. Springer.

2. R. Agrawal, T. Imielinski, and A. N. Swami. Mining Association Rules between Sets of Items in Large Databases. In P. Buneman and S. Jajodia, editors, *Proceedings of the ACM International Conference on Management of Data (SIGMOD'93)*, pp. 207–216, Washington, DC, May 1993. ACM Press.

3. R. Albert, H. Jeong, and A.-L. Barabási. The Diameter of the World Web Web. *Nature*, Vol. 401:pp. 130–131, September 1999.

4. J. Alspector, A. Kolez, and N. Karunanithi. Comparing Feature-Based and Clique-Based User Models for Movie Selection. In *Proceedings of the Third ACM Conference on Digital Libraries*, pp. 11–18, Pittsburgh, PA, June 1998. ACM Press.

5. E. André and T. Rist. From Adaptive Hypertext to Personalized Web Companions. *Communications of the ACM*, Vol. 45(5):pp. 43–46, May 2002.

6. L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley. Classes of Behavior of Small-World Networks. In *Proceedings of the National Academy of Science, USA*, Vol. 97, pp. 11149–11152, January 2000.

7. G. Adomavicius and A. Tuzhilin. User Profiling in Personalization Applications through Rule Discovery and Validation. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pp. 377–381, San Diego, CA, August 1999. ACM Press.

8. G. Adomavicius and A. Tuzhilin. Multidimensional Recommender Systems: A Data Warehousing Approach. In L. Fiege, G. Mühl, and U. G. Wilhelm, editors, *Second International Workshop on Electronic Commerce (WELCOM'01)*, Vol. 2232 of *Lecture Notes in Computer Science*, pp. 180–192, Heidelberg, Germany, November 2001. Springer-Verlag.

9. B. Amento, L. Terveen, and W. Hill. Does "Authority" Mean Quality? Predicting Expert Quality Ratings of Web Documents. In *Proceedings of the Twenty-third Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'00)*, pp. 296–303, Athens, Greece, July 2000. ACM Press.

10. C. C. Aggarwal, J. L. Wolf, K. Wu, and P. S. Yu. Horting Hatches an Egg: A Graph-Theoretic Approach to Collaborative Filtering. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pp. 201–212, San Diego, CA, August 1999. ACM Press.

11. C. Avery and R. Zeckhauser. Recommender Systems for Evaluating Computer Messages. *Communications of the ACM*, Vol. 40(3):pp. 88–89, March 1997.

12. A.-L. Barabási and R. Albert. Emergence of Scaling in Random Networks. *Science*, Vol. 286:pp. 509–512, October 1999.

13. P. Baudisch. Joining Collaborative and Content-based Filtering. In *Proceedings of the ACM CHI Workshop on Interacting with Recommender Systems*, Pittsburgh, PA, May 1999. ACM Press.

14. N. J. Belkin and W. B. Croft. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Communications of the ACM*, Vol. 35(12):pp. 29–38, December 1992.

15. M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review*, Vol. 37(4):pp. 573–595, January–February 1995.

16. N. J. Belkin. Helping People Find What They Don't Know. *Communications of the ACM*, Vol. 43(8):pp. 58–61, August 2000.

17. E. Bergman, editor. *Information Appliances and Beyond*. The Morgan Kaufmann Series on Interactive Technologies. Morgan Kaufmann, San Francisco, CA, 2000.

18. H. Berghel. Caustic Cookies. *Communications of the ACM*, Vol. 44(5):pp. 19–22, May 2001.

19. C. Basu and H. Hirsh. Using Multiple Information Sources for Recommendation. In *Proceedings of the Twenty-fourth Annual International ACM SIGIR Conference, Workshop on Recommender Systems*, New Orleans, LA, November 2001. ACM Press.

20. J. S. Breese, D. Heckerman, and C. Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43–52, Madison, WI, July 1998.

21. A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph Structure in the Web. In *Proceedings of the Ninth International World Wide Web Conference (WWW9)*, Amsterdam, Netherlands, May 2000.

22. E. Bloedorn, I. Mani, and T. R. MacMillan. Representational Issues in Machine Learning of User Profiles. In *AAAI Spring Symposium on Machine Learning in Information Access (MLIA)*, Stanford, CA, March 1996. AAAI Press.

23. S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, Brisbane, Australia, April 1998. Elsevier Science.

24. A. Broder. Exploring, Modeling, and Using the Web Graph. Keynote address to the Twenty-six Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'03), July–August 2003.

25. M. Balabanović and Y. Shoham. Fab: Content-Based, Collaborative Recommendation. *Communications of the ACM*, Vol. 40(3):pp. 66–72, March 1997.

26. R. Burke. Integrating Knowledge-Based and Collaborative Filtering Recommender Systems. In *Proceedings of the Workshop on Artificial Intelligence for Electronic Commerce*, pp. 69–72, Orlando, FL, July 1999. AAAI Press.

27. V. Bush. As We May Think. *The Atlantic Monthly*, Vol. 176(1):pp. 101–108, July 1945.

28. M. Claypool, D. Brown, P. Le, and M. Waseda. Inferring User Interest. *IEEE Internet Computing*, Vol. 5(6):pp. 32–39, November–December 2001.

29. J. M. Carlson and J. Doyle. Highly Optimized Tolerance: A Mechanism for Power Laws in Designed Systems. *Physical Review E*, Vol. 60(2):pp. 1412–1427, August 1999.

30. I. Cingil, A. Dogac, and A. Azgin. A Broader Approach to Personalization. *Communications of the ACM*, Vol. 43(8):pp. 136–141, August 2000.

31. S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the Web's Link Structure. *IEEE Computer*, Vol. 32(8):pp. 60–67, August 1999.

32. M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin. Combining Content-Based and Collaborative Filters in an Online Newspaper. In *Proceedings of the Twenty-second Annual International ACM SIGIR Conference, Workshop on Recommender Systems*, Berkeley, CA, August 1999. ACM Press.

33. J. M. Carroll and M. B. Rosson. Developing the Blacksburg Electronic Village. *Communications of the ACM*, Vol. 39(12):pp. 69–74, December 1996.

34. P. Denning. Electronic Junk. *Communications of the ACM*, Vol. 25(3):pp. 163–165, March 1982.

35. P. Erdøs and A. Rényi. On the Evolution of Random Graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, Vol. 5:pp. 17–61, 1960.

36. P. W. Foltz and S. T. Dumais. Personalized Information Delivery: An Analysis of Information Filtering Methods. *Communications of the ACM*, Vol. 35(12):pp. 51–60, December 1992.

37. Y. Fruend, R. Iyer, R. Schapire, and Y. Singer. An Efficient Boosting Algorithm for Combining Preferences. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 170–178, Madison, WI, July 1998. Morgan Kaufmann.

38. G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-Organization and Identification of Web Communities. *IEEE Computer*, Vol. 35(3):pp. 66–67, March 2002.

39. A. Grasso, M. Koch, and A. Rancati. Augmenting Recommender Systems by Embedding Interfaces into Practices. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work (GROUP'99)*, pp. 267–275, Phoenix, AZ, November 1999. ACM Press.

40. D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM*, Vol. 35(12):pp. 61–70, December 1992.

41. K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A Constant Time Collaborative Filtering Algorithm. Technical Report M00/41, Electronic Research Laboratory, University of California, Berkeley, August 2000.

42. J. Goecks and J. Shavlik. Learning Users' Interests by Unobtrusively Observing their Normal Behavior. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces (IUI'00)*, Observing User Behavior, pp. 129–132, New Orleans, LA, January 2000. ACM Press.

43. B. Hayes. Graph Theory in Practice: Part I. *American Scientist*, Vol. 88(1):pp. 9–13, January–February 2000.

44. B. Hayes. Graph Theory in Practice: Part II. *American Scientist*, Vol. 88(2):pp. 104–109, March–April 2000.

45. E. Housman and E. Kaskela. State of the Art in Selective Dissemination of Information. In *Proceedings of the IEEE Transaction on Engineering and Writing Speech*, pp. 100–112, Stanford, CA, March 1996. AAAI Press.

46. J. Herlocker, J. A. Konstan, and J. Riedl. Explaining Collaborative Filtering Recommendations. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'00)*, pp. 241–250, Philadelphia, PA, December 2000. ACM Press.

47. C. Hayes, P. Massa, P. Avesani, and P. Cunningham. An On-line Evaluation Framework for Recommender Systems. Technical Report TCD-CS-2002-19, Department of Computer Science, Trinity College Dublin, April 2002.

48. W. Hill and L. Terveen. Using Frequency-of-Mention in Public Conversations for Social Filtering. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pp. 106–112, Boston, MA, November 1996. ACM Press.

49. B. Krulwich and C. Burkley. Learning User Information Interests Through Extraction of Semantically Significant Phrases. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pp. 100–112, Stanford, CA, March 1996. AAAI Press.

50. J. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The Web as a Graph: Measurements, Models and Methods. In *Proceedings of the International Conference on Combinatorics and Computing*, 1999.

51. J. Kleinberg and S. Lawrence. The Structure of the Web. *Science*, Vol. 294:pp. 1849–1850, November 2001.

52. J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, Vol. 46(5):pp. 604–632, September 1999.

53. J. Kleinberg. Navigation in a Small World. *Nature*, Vol. 406:p. 845, August 2000.

54. J. Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *Proceedings of the Thirty-second ACM Symposium on Theory of Computing (STOC'00)*, pp. 163–170, Portland, OR, 2000. ACM Press.

55. J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying Collaborative Filtering to Usenet News. *Communications of the ACM*, Vol. 40(3):pp. 77–87, March 1997.

56. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for Emerging Cyber-Communities. In *Proceedings of the Eighth International World Wide Web Conference (WWW8)*, Toronto, Canada, May 1999.

57. H. Kautz, B. Selman, and M. Shah. Referral Web: Combining Social Networks and Collaborative Filtering. *Communications of the ACM*, Vol. 40(3):pp. 63–65, March 1997.

58. H. Kautz, B. Selman, and M. Shah. The Hidden Web. *AI Magazine*, Vol. 18(2):pp. 27–36, 1997.

59. G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item to Item Collaborative Filtering. *IEEE Internet Computing*, Vol. 7(1):pp. 76–80, January–February 2003.

60. S. Loeb and D. Terry. Information Filtering. *Communications of the ACM*, Vol. 35(12):pp. 26–28, December 1992.

61. C. Lynch. Personalization and Recommender Systems in the Larger Context: New Directions and Research Questions (Keynote Speech). In *Proceedings of the Joint DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries*, pp. 84–88, Dublin, Ireland, 2001.

62. M. D. Mulvenna, S. S. Anand, and A. G. Büchner. Personalization on the Net using Web Mining. *Communications of the ACM*, Vol. 43(8):pp. 122–125, August 2000.

63. F. Mintzer, G. W. Braudaway, F. P. Giordano, J. C. Lee, Karen A. Magerlein, S. D'Auria, A. Ribak, G. Shapir, F. Schiattarella, J. Tolva, and A. Zelenkov. Populating the Hermitage Museum's New Web Site. *Communications of the ACM*, Vol. 44(8):pp. 52–60, August 2001.

64. B. Mobashier, R. Cooley, and J. Srivastava. Automatic Personalization Based on Web Usage Mining. *Communications of the ACM*, Vol. 43(8):pp. 142–151, August 2000.

65. D. Maltz and K. Ehrlich. Pointing the Way: Active Collaborative Filtering. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'95)*, pp. 202–209, Denver, CO, May 1995. ACM Press.

66. S. Milgram. The Small World Problem. *Psychology Today*, Vol. 1(61):pp. 56–58, 1967.

67. B. J. Mirza. Jumping Connections: A Graph-Theoretic Model for Recommender Systems. Master's thesis, Department of Computer Science, Virginia Tech, February 2001. Available at http://scholar.lib.vt.edu/theses/available/etd-02282001-175040/.

68. B. J. Mirza, B. J. Keller, and N. Ramakrishnan. Studying Recommendation Algorithms by Graph Analysis. *Journal of Intelligent Information Systems*, Vol. 20(2):pp. 131–160, 2003.

69. J. Mostafa, S. Mukhopadhyay, W. Lam, and M. Palakal. A Multilevel Approach to Intelligent Information Filtering: Model, System, and Evaluation. *ACM Transactions on Information Systems*, Vol. 15(4):pp. 368–399, October 1997.

70. U. Manber, A. Patel, and J. Robinson. Experience with Personalization on Yahoo! *Communications of the ACM*, Vol. 43(8):pp. 35–39, August 2000.

71. R. Mooney and L. Roy. Content-Based Book Recommending Using Learning for Text Categorization. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pp. 195–204, San Antonio, TX, July 2000. ACM Press.

72. C. Nevill-Manning. The Biological Digital Library. *Communications of the ACM*, Vol. 44(5):pp. 41–42, May 2001.

73. F. Osareh. Bibliometrics, Citation Analysis and Co-Citation Analysis: A Review of Literature I. *Libri*, Vol. 46:pp. 149–158, 1996.

74. C. Papatheodorou. Machine Learning in User Modeling. *Lecture Notes in Computer Science*, Vol. 2049:pp. 286–294, September 2001.

75. M. J. Pazzani and D. Billsus. Learning and Revising User Profiles: The Identification of Interesting Web Sites. *Machine Learning*, Vol. 27(3):pp. 313–331, 1997.

76. M. Perkowitz and O. Etzioni. Adaptive Web Sites. *Communications of the ACM*, Vol. 43(8):pp. 152–158, August 2000.

77. M. Pazzani, J. Muramatsu, and D. Billsus. Syskill and Webert: Identifying Interesting Web Sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pp. 54–61, Portland, OR, August 1996. AAAI Press.

78. S. Perugini and N. Ramakrishnan. Personalizing Interactions with Information Systems. In M. V. Zelkowitz, editor, *Advances in Computers*, Vol. 57: Information Repositories, pp. 323–382. Academic Press, September 2003. **Invited contribution**.

79. D. Riecken. Personalized Views of Personalization. *Communications of the ACM*, Vol. 43(8):pp. 27–28, August 2000.

80. J. Riedl. Personalization and Privacy. *IEEE Internet Computing*, Vol. 5(6):pp. 29–31, November-December 2001.

81. P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'94)*, pp. 175–186, Chapel Hill, NC, October 1994. ACM Press.

82. N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis. Privacy Risks in Recommender Systems. *IEEE Internet Computing*, Vol. 5(6):pp. 54–62, November–December 2001.

83. S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ, 1995.

84. J. J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.

85. J. Rcuker and M. J. Polano. Siteseer: Personalized Navigation for the Web. *Communications of the ACM*, Vol. 40(3):pp. 73–75, March 1997.

86. P. Resnick and H. R. Varian. Recommender Systems. *Communications of the ACM*, Vol. 40(3):pp. 56–58, March 1997.

87. P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation Systems. *Communications of the ACM*, Vol. 43(12):pp. 45–48, December 2000.

88. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 1998.

89. S. Srinivasan and E. Brown. Is Speech Recognition Becoming Mainstream? *IEEE Computer*, Vol. 35(4):pp. 38–41, April 2002.

90. J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, Vol. 1(2):pp. 12–23, January 2000.

91. U. Shardanand. *Social Information Filtering for Music Recommendation*. Ph.D. dissertation, Massachusetts Institute of Technology, 1994.

92. B. Shneiderman. Designing Trust into Online Experiences. *Communications of the ACM*, Vol. 43(12):pp. 57–59, December 2000.

93. B. Sarwar, J. Konstan, J. Borchers, A. Herlocker, J. Miller, and J. Riedl. Using Filtering Agents to Improve Prediction Quality in the GroupLens Research Collaborative Filtering System. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'98)*, pp. 345–354, Seattle, WA, November 1998. ACM Press.

94. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of Recommendation Algorithms for E-Commerce. In *Proceedings of the Second ACM Conference on Electronic Commerce*, pp. 158–167, Minneapolis, MN, 2000. ACM Press.

95. J. B. Schafer, J. A. Konstan, and J. Riedl. Recommender Systems in E-Commerce. In *Proceedings of the First ACM Conference on Electronic Commerce*, pp. 158–166, Denver, CO, November 1999. ACM Press.

96. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Computer Science Series. McGraw-Hill, New York, 1983.

97. U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'95)*, pp. 210–217, Denver, CO, May 1995. ACM Press.

98. I. Soboroff and C. Nicholas. Combining Content and Collaboration in Text Filtering. In *Proceedings of the IJCAI'99 Workshop on Machining Learning in Information Filtering*, pp. 86–91, Stockholm, Sweden, August 1999.

99. M. Spiliopoulou. Web Usage Mining for Web Site Evaluation. *Communications of the ACM*, Vol. 43(8):pp. 127–134, August 2000.

100. R. Sinha and K. Swearingen. Comparing Recommendations Made by Online Systems and Friends. In *Proceedings of the Joint DELOS-NSF Workshop on Personalisation and Recommender Systems in Digital Libraries*, pp. 73–78, Dublin, Ireland, June 2001.

101. K. Swearingen and R. Sinha. Beyond Algorithms: An HCI perspective on Recommender Systems. In *Proceedings of the Twenty-fourth Annual International ACM SIGIR Conference, Workshop on Recommender Systems*, New Orleans, LA, November 2001. ACM Press.

102. R. Sinha and K. Swearingen. The Role of Transparency in Recommender Systems. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'02)*, pp. 830–831, Minneapolis, MN, April 2002. ACM Press.

103. K. Swearingen and R. Sinha. Interaction Design for Recommender Systems. In *Proceedings of the Conference on Designing Interactive Systems (DIS'02)*, London, England, June 2002. ACM Press.

104. G. W. Stewart. The Decomposition Approach To Matrix Computation. *IEEE/AIP Computing in Science and Engineering*, Vol. 2(1):pp. 50–58, January–February 2000.

105. C. Shapiro and H. R. Varian. *Information Rules: A Strategic Guide to the Network Economy*. Harvard Business School Press, November 1999.

106. M. F. Schwartz and D. C. M. Wood. Discovering Shared Interests Using Graph Analysis. *Communications of the ACM*, Vol. 36(8):pp. 78–89, August 1993.

107. G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, Vol. 18(11):pp. 613–620, November 1975.

108. M. P. Singh, B. Yu, and M. Venkatraman. Community-Based Service Location. *Communications of the ACM*, Vol. 44(4):pp. 49–54, April 2001.

109. L. Terveen and W. Hill. Human-Computer Collaboration in Recommender Systems. In J. M. Carroll, editor, *Human-Computer Interaction in the New Millennium*, Chapter 22. Addison-Wesley, 2002.

110. L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter. PHOAKS: A System for Sharing Recommendations. *Communications of the ACM*, Vol. 40(3):pp. 59–62, March 1997.

111. *ACM Transactions on Computer-Human Interaction*, 2004. Special issue on recommender system interfaces: theory and practice. To appear.

112. D. J. Watts. Kevin Bacon, the Small-World, and Why It All Matters. *Santa Fe Institute Bulletin*, Vol. 14(2), 1999.

113. B. Wellman. Computer Networks As Social Networks. *Science*, Vol. 293:pp. 2031–2034, September 2001.

114. S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, New York, 1994.

115. S. Wasserman and J. Galaskiewicz, editors. *Advances in Social Network Analysis: Research In The Social And Behavioral Sciences*. Sage, Thousand Oaks, CA, 1994.

116. A. Wexelblat and P. Maes. Footprints: History-Rich Tools for Information Foraging. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'99)*, pp. 270–277, Pittsburgh, PA, May 1999. ACM Press.

117. G. I. Webb, M. J. Pazzani, and D. Billsus. Machine Learning for User Modeling. *User Modeling and User-Adapted Interaction*, Vol. 11:pp. 19–29, 2001.

118. D. J. Watts and S. Strogatz. Collective Dynamics of 'Small-World' Networks. *Nature*, Vol. 393:pp. 440–442, June 1998.

119. T. W. Yan and H. García-Molina. The SIFT Information Dissemination System. *ACM Transactions on Database Systems*, Vol. 24(4):pp. 529–565, December 1999.

120. J. Zimmerman and K. Kurapati. Exposing Profiles to Build Trust in a Recommender. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'02)*, pp. 608–609, Minneapolis, MN, April 2002. ACM Press.