Studying Recommendation Algorithms
by Graph Analysis

Batul J. Mirza
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061
bmirza@csgrad.cs.vt.edu

Benjamin J. Keller
Department of Computer Science
Eastern Michigan University
Ypsilanti, MI 48197
Ben.Keller@emich.edu

Naren Ramakrishnan
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061
naren@cs.vt.edu

Abstract

We present a novel framework for studying recommendation algorithms in terms of the 'jumps' that they make to connect people to artifacts. This approach emphasizes reachability via an algorithm within the implicit graph structure underlying a recommender dataset and allows us to consider questions relating algorithmic parameters to properties of the datasets. For instance, given a particular algorithm 'jump,' what is the average path length from a person to an artifact? Or, what choices of minimum ratings and jumps maintain a connected graph? We illustrate the approach with a common jump called the 'hammock' using movie recommender datasets.

Keywords: Recommender Systems, Collaborative Filtering, Information System Evaluation, Random Graphs.

1 Introduction

Recommender systems [41] constitute one of the fastest growing segments of the Internet economy today. They help reduce information overload and provide customized information access for targeted domains. Building and deploying recommender systems has matured into a fertile business activity, with benefits in retaining customers and enhancing revenues. Elements of the recommender landscape include customized search engines, handcrafted content indices, personalized shopping agents on e-commerce sites, and news-on-demand services. The scope of such personalization thus extends to many different forms of information content and delivery, not just web pages. The underlying algorithms and techniques, in turn, range from simple keyword matching of consumer profiles, collaborative filtering, to more sophisticated forms of data mining, such as clustering web server logs.

Recommendation is often viewed as a system involving two modes (typically people and artifacts, such as movies and books) and has been studied in domains that focus on harnessing online information resources, information aggregation, social schemes for decision making, and user interfaces. A recurring theme among many of these applications is that recommendation is implicitly cast as a task of learning mappings (from people to recommended artifacts, for example) or of filling in entries to missing cells in a matrix (of consumer preferences, for example). Consequently, recommendation algorithms are evaluated by the accuracies of their predicted ratings. We approach recommendation from a different but complementary perspective of considering the connections that are made.

1.1 Motivating Scenarios

We describe three scenarios involving recommender system design to motivate the ideas presented in this paper.

- *Scenario 1:* A small town bookstore is designing a recommender system to provide targeted personalization for its customers. Transactional data, from purchases cataloged over three years, is available. The store is not interested in providing specific recommendations of books, but is keen on using its system as a means of introducing customers to one another and encouraging them to form reading groups. It would like to bring sufficiently concerted groups of people together, based on commonality of interests. Too many people in a group would imply a diffusion of interests; modeling reading habits too narrowly might imply that some people cannot be matched with anybody. How can the store relate commonality of interests to the sizes of clusters of people that are brought together?

- *Scenario 2:* An e-commerce site, specializing in books, CDs, and movie videos, is installing a recommendation service. The designers are acutely aware that people buy and rate their different categories of products in qualitatively different ways. For example, movie ratings follow a *hits-buffs* distribution: some people (the buffs) see/rate almost all movies, and some movies (the hits) are seen/rated by almost all people. Music CD ratings are known to be more clustered, with hits-buffs distributions visible only within specific genres (like 'western classical'). Connections between different genres are often weak, compared to connections within a genre. How can the designers reason about and visualize the structure of these diverse recommendation spaces, to allow them to custom-build their recommendation algorithms?

- *Scenario 3:* An online financial firm is investing in a recommendation service and is requiring each of its members to rate at least κ products of their own choice to ensure that there are enough overlaps among ratings. The company's research indicates that people's ratings typically follow power-law distributions. Furthermore, the company's marketers have decided that recommendations of ratings can be explainably transferred from one person to another if they have at least 6 ratings in common. Given these statistics and design constraints, what value of κ should be set by the company to ensure that every person (and every product) is reachable by its recommendation service?

The common theme among these applications is that they emphasize many important aspects of a recommender system other than predictive accuracy: its role as an indirect way of bringing people together, its signature pattern of making connections, and the explainability of its recommendations. To address the questions raised by considering these aspects of recommendation, we propose a framework based on a mathematical model of the social network implicit in recommendation. This framework allows a more direct approach to reasoning about recommendation algorithms and their relationship to the recommendation patterns of users. We effectively ignore the issue of predictive accuracy, and so the framework is a complement to approaches based on field studies.

The defined framework depends only on connections between people and artifacts, and so merely requires that we can capture the effect of a recommendation algorithm in such terms. This does not mean that the framework only applies to collaborative filtering where user-artifact relationships are explicitly collected. Other approaches that implicitly connect a user to an artifact can also be studied in this framework (e.g., by web page visits, using co-occurrence patterns). Algorithms that use direct measures of the similarity of the attributes of users (e.g., age) can also be modeled, although we do not describe how in this paper.

Our framework allows a novel strategy for characterizing a recommender algorithm in terms of the connections that it makes. We explore the foundations of this strategy for a generalization of the connections made by most common (collaborative filtering) algorithms. In particular, we look at characteristics of the social network graph induced by this algorithm, relating to whether recommendations can be made (connectivity of the social network) and how much effort is required (path lengths in the social network).

1.2 Reader's Guide

Section 2 surveys current research and motivates the need for a new approach to analyzing algorithms for recommender systems. Section 3 introduces the jumping connections framework and develops a mathematical model based on random graph theory. Section 4 provides experimental results for one particular way of jumping on two application datasets. Issues related to interpretation of results from our model are also presented here. Finally, Section 5 identifies some opportunities for future research.

2 Characterizing Recommendation Algorithms

Most current research efforts cast recommendation as a specialized task of information retrieval/filtering or as a task of function approximation/learning mappings [2, 7, 8, 19, 20, 23, 24, 27, 31, 39, 46, 47, 48, 50, 51, 53]. Even approaches that focus on clustering view

clustering primarily as a pre-processing step for functional modeling [30], or as a technique to ensure scalability [37, 47] or to overcome sparsity of ratings [54]. This emphasis on functional modeling and retrieval has influenced evaluation criteria for recommender systems.

Traditional information retrieval evaluation metrics such as precision and recall have been applied toward recommender systems involving content-based design. Ideas such as cross-validation on an unseen test set have been used to evaluate mappings from people to artifacts, especially in collaborative filtering recommender systems. Such approaches miss many desirable aspects of the recommendation process, namely:

- **Recommendation is an indirect way of bringing people together.** As we will discuss recommendation algorithms, especially collaborative filtering, exploit connections between users and artifacts. Social network theory [55] can be used to model such a recommendation system of people versus artifacts as an *affiliation network* and distinguishes between a *primary mode* (e.g., people) and a *secondary mode* (e.g., movies), where a *mode* refers to a distinct set of entities that have similar attributes [55]. The purpose of the secondary mode is viewed as serving to bring entities of the primary mode together (i.e., it isn't treated as a *first-class* mode).

- **Recommendation, as a process, should emphasize modeling connections from people to artifacts, besides predicting ratings for artifacts.** In many situations, users would like to request recommendations purely based on local and global constraints on the nature of the specific connections explored. Functional modeling techniques are inadequate because they embed the task of learning a mapping from people to predicted values of artifacts in a general-purpose learning system such as neural networks or Bayesian classification [10]. A notable exception is the work by Hofmann and Puzicha [25] which allows the incorporation of constraints in the form of aspect models involving a latent variable.

- **Recommendations should be explainable and believable.** The explanations should be made in terms and constructs that are natural to the user/application domain. It is nearly impossible to convince the user of the quality of a recommendation obtained by black-box techniques such as neural networks. Furthermore, it is well recognized that "users are more satisfied with a system that produces [bad recommendations] for reasons that seem to make sense to them, than they are with a system that produces [bad recommendations] for seemingly stupid reasons" [44].

- **Recommendations are not delivered in isolation, but in the context of an implicit/explicit social network.** In a recommender system, the rating patterns of people on artifacts induce an implicit social network and influence the connectivities in this network. Little study has been done to understand how such rating patterns influence recommendations and how they can be advantageously exploited.

Our approach in this paper is to study recommendation algorithms using ideas from graph analysis. In the next section, we will show how our viewpoint addresses each of the above aspects, by providing novel metrics. The basic idea is to begin with data that can be modeled as a network and attempt to infer useful knowledge from the vertices and edges of the graph. Vertices represent entities in the domain (e.g., people, movies), and edges represent the relationships between entities (e.g., the act of a person viewing a particular movie).

2.1 Related Research

The idea of graph analysis as a basis to study information networks has a long tradition; one of the earliest pertinent studies is Schwartz and Wood [49]. The authors describe the use of graph-theoretic notions such as cliques, connected components, cores, clustering, average path distances, and the inducement of secondary graphs. The focus of the study was to model shared interests among a web of people, using email messages as connections. Such link analysis has been used to extract information in many areas such as in web search engines [28], in exploration of associations among criminals [42], and in the field of medicine [52]. With the emergence of the web as a large scale graph, interest in information networks has recently exploded [1, 11, 12, 14, 17, 26, 28, 29, 32, 38, 40, 56].

Most graph-based algorithms for information networks can be studied in terms of (i) the graph modeling (e.g., what are the modes?, how do they relate to the information domain?), and (ii) the structures/operations that are mined/conducted on the graph. One of the most celebrated examples of graph analysis arises in search engines that exploit both link information and textual content. The Google search engine uses the web's link structure in addition to the anchor text, as a factor in ranking pages based on the pages that (hyper)link to the given page [11]. Google essentially models a one-mode directed graph (of web pages) and uses measures involving principal components to ascertain 'page ranks.' Jon Kleinberg's HITS (Hyperlink-Induced Topic Search) algorithm goes a step further by viewing the one-mode web graph as actually comprising two modes (called hubs and authorities) [28]. A hub is a node primarily with edges to authorities, and so a good hub has links to many authorities. A good authority is a page that is linked to by many hubs. Starting with a specific search query, HITS performs a text-based search to seed an initial set of results. An iterative relaxation algorithm then assigns hub and authority weights using a matrix power iteration. Empirical results show that remarkably authoritative results are obtained for search queries. The CLEVER search engine is built primarily on top of the basic HITS algorithm [14].

The use of link analysis in recommender systems was highlighted by the "referral chaining" technique of the ReferralWeb project [26]. The idea is to use the co-occurrence of names in any of the documents available on the web to detect the existence of direct relationships between people and discover a social network. The underlying assumption is that people with similar interests swarm in the same circles to discover collaborators [38].

The exploration of link analysis in social structures has led to several new avenues of research, most notably small-world networks. Small-world networks are highly clustered but relatively sparse networks with small average length. An example is the folklore notion of six degrees of separation separating any two people in our universe: the phenomenon where a person can discover a link to any other random person through a chain of at most six acquaintances. A small-world network is sufficiently clustered so that most second neighbors of a node X are also neighbors of X (a typical ratio would be 80%). On the other hand, the average distance between any two nodes in the graph is comparable to the low characteristic path length of a random graph. Until recently, a mathematical characterization of such small-world networks has proven elusive. Watts and Strogatz [56] provide the first such characterization of small-world networks in the form of a graph generation model.

In this model, Watts and Strogatz use a regular wreath network with n nodes, and k edges per node (to its nearest neighbors) as a starting point for the design. A small fraction
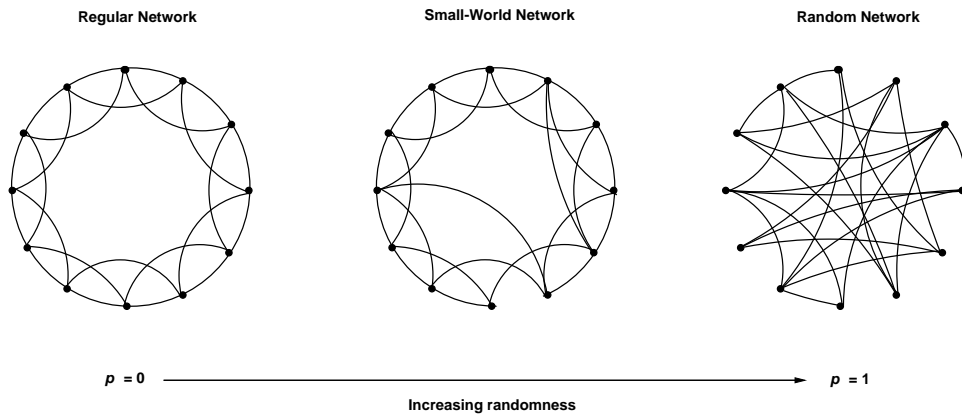
Figure 1: Generation of a small-world network by random rewiring from a regular wreath network. Figure adapted from [56].
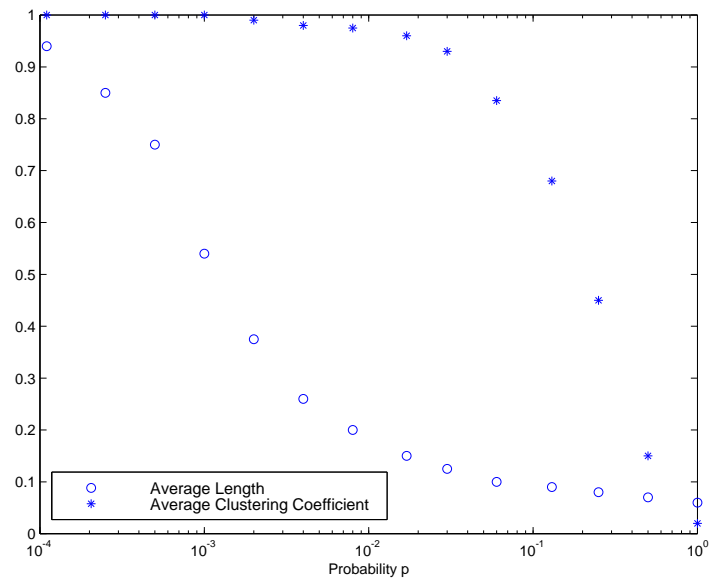

Figure 2: Average path length and clustering coefficient versus the rewiring probability p (from [56]). All measurements are scaled w.r.t. the values at $p = 0$.

of the edges are then randomly rewired to arbitrary points on the network. A full rewiring (probability $p = 1$) leads to a completely random graph, while $p = 0$ corresponds to the (original) wreath (Fig. 1). The starting point in the figure is a regular wreath topology of 12 nodes with every node connected to its four nearest neighbors. This structure has a high characteristic path length and high clustering coefficient. The average length is the mean of the shortest path lengths over all pairs of nodes. The clustering coefficient is determined by first computing the local neighborhood of every node. The number of edges in this neighborhood as a fraction of the total possible number of edges denotes the extent of the neighborhood being a clique. This factor is averaged over all nodes to determine the clustering coefficient. The other extreme in Fig. 1 is a random network with a low characteristic path length and almost no clustering. The small-world network, an interpolation between the two, has the low characteristic path length (of a random network), and retains the high clustering coefficient (of the wreath). Measuring properties such as average length and clustering coefficient in the region $0 \leq p \leq 1$ produces surprising results (see Fig. 2).

As shown in Fig. 2, only a very small fraction of edges need to be rewired to bring the length down to random graph limits, and yet the clustering coefficient is high. On closer inspection, it is easy to see why this should be true. Even for small values of p (e.g., 0.1), the result of introducing edges between distantly separated nodes reduces not only the distance between these nodes but also the distances between the neighbors of those nodes, and so on (these reduced paths between distant nodes are called *shortcuts*). The introduction of these edges further leads to a rapid decrease in the average length of the network, but the clustering coefficient remains almost unchanged. Thus, small-world networks fall in between regular and random networks, having the small average lengths of random networks but high clustering coefficients akin to regular networks.

While the Watts-Strogatz model describes how small-world networks can be formed, it does not explain how people are adept at actually finding short paths through such networks in a decentralized fashion. Kleinberg addresses precisely this issue and proves that this is not possible in the family of one-dimensional Watts-Strogatz networks [29]. Embedding the notion of random rewiring in a two-dimensional lattice leads to one unique model for which such decentralization is effective.

The small-world network concept has implications for a variety of domains. Watts and Strogatz simulate the spread of an infectious disease in a small-world network [56]. Adamic shows that the world wide web is a small-world network and suggests that search engines capable of exploiting this fact can be more effective in hyperlink modeling, crawling, and finding authoritative sources [1].

Besides the Watts-Strogatz model, a variety of models from graph theory are available and can be used to analyze information networks. Kumar et al. [32] highlight the use of traditional random graph models to confirm the existence of properties such as cores and connected components in the web. In particular, they characterize the distributions of web page degrees and show that they are well approximated by power laws. Finally, they perform a study similar to Schwartz and Wood [49] to find cybercommunities on the web. Flake et al. [17] provide a max-flow, min-cut algorithm to identify cybercommunities. They also provide a focused crawling strategy to approximate such communities. Broder et al. [12] perform a more detailed mapping of the web and demonstrate that it has a bow-tie structure, which consists of a strongly connected component, as well as nodes that link into but are not linked from the strongly connected component, and nodes that are linked from but do

not link to the strongly connected component. Pirolli et al. [40] use ideas from spreading activation theory to subsume link analysis, content-based modeling, and usage patterns.

A final thread of research, while not centered on information networks, emphasizes the modeling of problems and applications in ways that make them amenable to graph-based analyses. A good example in this category is the approach of Gibson et al. [18] for mining categorical datasets.

While many of these ideas, especially link analysis, have found their way into recommender systems, they have been primarily viewed as mechanisms to mine or model structures. In this paper, we show how ideas from graph analysis can actually serve to characterize recommender algorithms.

3 Graph Analysis

In the previous section, we identified four pertinent aspects of recommendation:

- bringing people together by connections; in

- an underlying graph-theoretic structure;

- explainability; and

- modeling of social networks.

To address these four aspects, we develop a novel way to study algorithms for recommender systems. Algorithms are distinguished, not by the predicted ratings of services/artifacts they produce, but by the combinations of people and artifacts that they bring together. Two algorithms are considered equivalent if they bring together identical sets of nodes regardless of whether they work in qualitatively different ways. Our emphasis is on the role of a recommender system as a mechanism for bridging entities in a social network. We refer to this approach of studying recommendation as *jumping connections*.

Notice that the framework does not emphasize how the recommendation is actually made, or the information that an algorithm uses to make connections (e.g., does it rely on others' ratings, on content-based features, or both?). In addition, we make no claims about the recommendations being better or that they will be better received. Our metrics, hence, will not lead a designer to directly conclude that an algorithm A is more accurate than an algorithm B; such conclusions can only be made through a field evaluation (involving feedback and reactions from users) or via survey/interview procedures. By restricting its scope to exclude the actual aspect of making ratings and predictions, the jumping connections framework provides a systematic and rigorous way to study recommender systems.

Of course, the choice of how to jump connections will be driven by the (often conflicting) desire to reach almost every node in the graph (i.e., recommend every product for somebody, or recommend some product for everybody) and the strength of the jumps enjoyed when two nodes are brought together. The conflict between these goals can be explicitly expressed in our framework.

It should be emphasized that our model doesn't imply that algorithms only exploit local structure of the recommendation dataset. Any mechanism — local or global — could be

used to jump connections. In fact, it is not even necessary that algorithms employ graph-theoretic notions to make connections. Our framework only requires a boolean test to see if two nodes are brought together.

Notice also that when an algorithm brings together person X and artifact Y, it could imply either a positive recommendation or a negative one. Such differences are, again, not captured by our framework unless the mechanism for making connections restricts its jumps, for instance, to only those artifacts for which ratings satisfy some threshold. In other words, thresholds for making recommendations could be abstracted into the mechanism for jumping.

Jumping connections satisfies all the aspects outlined in the previous section. It involves a social-network model, and thus, emphasizes connections rather than prediction. The nature of connections jumped also aids in explaining the recommendations. The graph-theoretic nature of jumping connections allows the use of mathematical models (such as random graphs) to analyze the properties of the social networks in which recommender algorithms operate.

3.1 The Jumping Connections Construction

We now develop the framework of jumping connections. We use concepts from a movie recommender system to provide the intuition; this does not restrict the range of applicability of jumping connections and is introduced here only for ease of presentation.

A *recommender dataset* $\mathcal{R}$ consists of the ratings by a group of people of movies in a collection. The ratings could in fact be viewings, preferences, or other constraints on movie recommendations. Such a dataset can be represented as a bipartite graph $G = (P \cup M, E)$ where P is the set of people, M is the set of movies, and the edges in E represent the ratings of movies. We denote the number of people by $N_P = |P|$, and the number of movies as $N_M = |M|$.

We can view the set M as a secondary mode that helps make connections — or jumps — between members of P. A *jump* is a function $\mathcal{J} : \mathcal{R} \mapsto S, S \subseteq P \times P$ that takes as input a recommender dataset $\mathcal{R}$ and returns a set of (unordered) pairs of elements of P. Intuitively, this means that the two nodes described in a given pair can be reached from one another by a single jump. Notice that this definition does not prescribe how the mapping should be performed, or whether it should use all the information present in $\mathcal{R}$. We also make the assumption that jumps can be composed in the following sense: if node B can be reached from A in one jump, and C can be reached from B in one jump, then C is reachable from A in two jumps. The simplest jump is the *skip*, which connects two members in P if they have at least one movie in common.

A jump induces a graph called a social network graph. The *social network graph* of a recommender dataset $\mathcal{R}$ induced by a given jump $\mathcal{J}$ is a unipartite undirected graph $G_s = (P, E_s)$, where the edges are given by $E_s = \mathcal{J}(\mathcal{R})$. Notice that the induced graph could be disconnected based on the strictness of the jump function. Figure 3 (b) shows the social network graph induced from the example in Figure 3 (a) using a skip jump.

We view a recommender system as exploiting the social connections (the jumps) that bring together a person with other people who have rated an artifact of (potential) interest. To model this, we view the unipartite social network of people as a directed graph and reattach movies (seen by each person) such that every movie is a sink (reinforcing its role as a secondary mode). The shortest paths from a person to a movie in this graph can then be
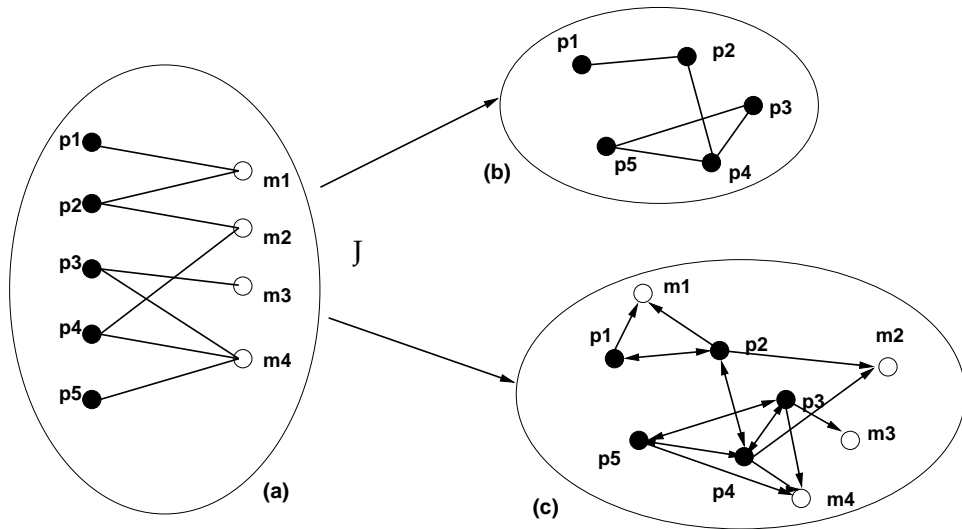
Figure 3: Illustration of the *skip* jump. (a) bipartite graph of people and movies. (b) Social network graph, and (c) recommender graph.

used to provide the basis for recommendations. We refer to a graph induced in this fashion as a *recommender graph* (Figure 3 (c)). Since the outdegree of every movie node is fixed at zero, paths through the graph are from people to movies (through more people, if necessary).

The recommender graph of a recommender dataset $\mathcal{R}$ induced by a given jump function $\mathcal{J}$ is a directed graph $G_r = (P \cup M, E_{sd} \cup E_{md})$, where E_{sd} is an ordered set of pairs, listing every pair from $\mathcal{J}(\mathcal{R})$ in both directions, and E_{md} is an ordered set of pairs, listing every pair from E in the direction pointing to the movie mode.

Assuming that the jump construction does not cause G_r to be disconnected, the portion of G_r containing only people is its strongest component: every person is connected to every other person. The movies constitute vertices which can be reached from the strongest component, but from which it is not possible to reach the strongest component (or any other node, for that matter). Thus, G_r can be viewed as a 'half bow-tie,' (Figure 4) as contrasted to the full bow-tie nature of the web, observed by Broder et al. [12]. The circular portion in the figure depicts the strongly connected component derived from G_s. Links out of this portion of the graph are from people nodes and go to sinks, which are movies.

3.2 Basic Approach

The skip jump is admittedly a simplistic way to model a recommendation algorithm and is introduced here only for illustration. A realistic recommendation algorithm would be more careful in bringing people nodes together; it might consider aspects such as commonality of rated artifacts, agreement among ratings, correlations, and other metrics of statistical similarity. There are thus many ways of inducing the social network graph and the recommender graph.

Our basic approach is to think of the operation of a given recommendation algorithm as performing a jump and study this jump algorithmically using tools from random graph theory. Our specific contributions in this paper are the emphasis on graph analysis as a way to
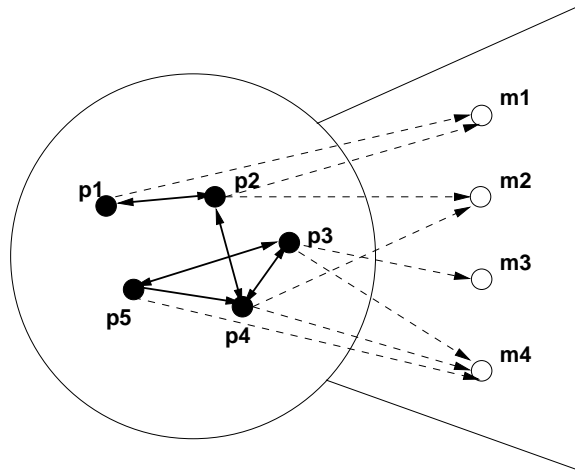
Figure 4: The jumping connections construction produces a half bow-tie graph G_r.
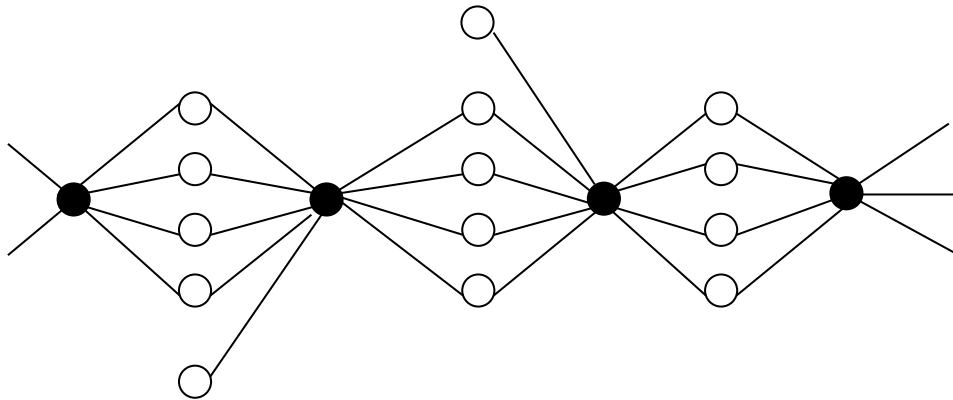

Figure 5: A path of hammock jumps, with a hammock width $w = 4$.

study recommendation algorithms and the use of random graph models to predict properties of interest about the social network and recommender graphs. The mathematical machinery brought to bear upon this problem is the work of Newman, Strogatz, and Watts [36]; the use of these formulas for studying recommendation algorithms is novel here.

For want of space, we are unable to study all or even a considerable number of jumps from the recommender systems literature. Instead, we focus on one jump called the hammock jump and describe the full extent of the analyses that can be conducted within our proposed framework. A more comprehensive list of jumps defined by different algorithms is explored by Mirza [35].

3.3   Hammocks

A hammock jump brings two people together in G_s if they have at least w movies in common in $\mathcal{R}$. Formally, a pair (p_1, p_2) is in $\mathcal{J}(\mathcal{R})$ whenever there is a set $M_{(p_1,p_2)}$ of w movies such that there is an edge from p_1 and p_2 to each element of $M_{(p_1,p_2)}$. The number w of common artifacts is called the hammock width. Figure 5 illustrates a sequence (or *hammock path*) of hammocks.

There is some consensus in the community that hammocks are fundamental in recommender algorithms since they represent commonality of ratings. It is our hypothesis that hammocks are fundamental to all recommender system jumps. Early recommendation projects such as GroupLens [31], LikeMinds [33], and Firefly [50] can be viewed as employing (simple versions of) hammock jumps involving at most one intermediate person.

The horting algorithm of Aggarwal et al. [2] extends this idea to a sequence of hammock jumps. Two relations — horting and predictability — are used as the basis for a jump. A person p_1 *horts* person p_2 if the ratings they have in common are a sufficiently large subset of the ratings of p_1. A person *predicts* another if they have a reverse horting relationship, and if there is a linear transformation between their ratings. The algorithm first finds shortest paths of hammocks that relate to predictability and then propagates ratings using the linear transformations. The implementation described by Aggarwal et al. [2] uses a bound on the length of the path.

There are a number of interesting algorithmic questions that can be studied. First, since considering more common ratings can be beneficial (see [23] for approaches) having a wider hammock could be better (this is not exactly true, when correlations between ratings are considered [23]). Second, many recommender systems require a minimum number κ of ratings before the user may use the system, to prevent *free-riding* on recommendations [5]. What is a good value for κ? And, third what is the hammock diameter or how far would we have to traverse to reach everyone in the social network graph? We begin looking at these questions in the next section.

3.4   Random Graph Models

Our goal is to be able to answer questions about hammock width, minimum ratings, and path length in a typical graph. The approach we take is to use a model of random graphs adapted from the work of Newman, Strogatz, and Watts [36]. This model, while having limitations, is the best-fit of existing models, and as we shall see, provides imprecise but descriptive results.

A recommender dataset $\mathcal{R}$ can be characterized by the number of ratings that each person makes, and the number of ratings that each artifact receives. These values correspond to the degree distributions in the bipartite rating graph for $\mathcal{R}$. These counts are relatively easy to obtain from a dataset and so could be used in the analysis of recommender algorithms. Therefore, we would like to be able to characterize a random bipartite graph using particular degree distributions. This requirement means that the more common random graph models (e.g., [15]) are not appropriate, since they assume that edges occur with equal probability. On the other hand, a model recently proposed by Aiello, Chung, and Lu [3] is based on a power-law distribution, similar to characteristics observed of actual recommendation datasets (see next section). But again this model is not directly parameterized by the degree distribution. The Newman-Strogatz-Watts model is the only (known) model that characterizes a family of graphs in terms of degree distributions.

From the original bipartite graph $G = (P \cup M, E)$ for $\mathcal{R}$ we develop two models, one for the social network graph G_s and one for the recommender graph G_r.

3.5 Modeling the Social Network Graph

Recall that the social network graph $G_s = (P, E_s)$ is undirected and E_s is induced by a jump function $\mathcal{J}$ on $\mathcal{R}$. The Newman-Strogatz-Watts model works by characterizing the degree distribution of the vertices, and then using that to compute the probability of arriving at a node. Together they describe a random process of following a path through a graph, and allow computations of the length of paths. Here we only discuss the equations that are used, and not the details of their derivation. The application of these equations to these graphs is outlined by Mirza [35] and is based on the derivation by Newman et al. [36].

We describe the social network graph G_s by the probability that a vertex has a particular degree. This is expressed as a generating function $G_0(x)$

$$G_0(x) = \sum_{k=0}^{\infty} p_k x^k,$$

where p_k is the probability that a randomly chosen vertex in G_s has degree k. This function must satisfy the property that

$$G_0(1) = \sum_{k=0}^{\infty} p_k = 1.$$

To obtain an expression that describes the typical length of a path, we can consider how many steps we need to go from a node to be able to get to every other node in the graph. To do this we can use the number of neighbors k steps away. For a randomly chosen vertex in this graph, $G_0(x)$ gives us the distribution of the immediate neighbors of that vertex. So, we can compute the average number of vertices z_1 one edge away from a vertex as the average degree z:

$$z_1 = z = \sum_k k p_k = G_0'(1)$$

The number of neighbors two steps away is given by

$$z_2 = \sum_k k p_k \frac{1}{z} \sum_k k(k-1) p_k$$

It turns out (see [36] for details) that the number of neighbors m steps away is given in terms of these two quantities:

$$z_m = \left(\frac{z_2}{z_1}\right)^{m-1} z_1$$

The path length l_{pp} we are interested in is the one that is big enough to reach all of the N_P elements of P, and so l_{pp} should satisfy the equation

$$1 + \sum_{m=1}^{l_{pp}} z_m = N_P$$

where the constant 1 counts the initial vertex. Using this equation, it can be shown that the typical length from one node to another in G_s is

$$l_{pp} = \frac{log[(N_P - 1)(z_2 - z_1) + z_1^2] - log[z_1^2]}{log[z_2/z_1]} \tag{1}$$

We use this formula as our primary means of computing the distances between pairs of people in G_s in the empirical evaluation in the next section. Since we use actual datasets, we can compute p_k as the fraction of vertices in the graph having degree k.

3.6 Modeling the Recommender Graph

The recommender graph $G_r = (P \cup M, E_{sd} \cup E_{md})$ is directed, and hence Newman, Strogatz, and Watts [36] capture both indegrees and outdegrees in a generating function:

$$G(x, y) = \sum_{j=0,k=0}^{j=\infty,k=\infty} p_{jk} x^j y^k,$$

where p_{jk} is the probability that a randomly chosen vertex has indegree j and outdegree k.

From the jumping connections construction, we know that movie vertices have outdegree 0 (the converse is not true, vertices with outdegree 0 could be people nodes isolated as a result of a severe jump constraint). Notice also that by using the joint distribution p_{ij}, independence of the indegree and outdegree distributions is *not* implied. We show in the next section that this feature is very useful. In addition, the average number of arcs entering (or leaving) a vertex is zero. And, so

$$\sum_{jk}(j - k)p_{jk} = \sum_{jk}(k - j)p_{jk} = 0.$$

We arrive at new expressions for z_1 and z_2 [35]:

$$z_1 = \sum_{jk} k p_{jk}.$$

$$z_2 = \sum_{jk} jk p_{jk}.$$

The average path length l_r can be calculated as before:

$$l_r = \frac{log[(N_P + N_M - 1)(z_2 - z_1) + z_1^2] - log[z_1^2]}{log[z_2/z_1]}, \tag{2}$$

where $N_P + N_M$ is the size of the recommender graph G_r (assuming that the graph is one giant component), with N_M denoting the number of movies. The length l_r includes paths from people to movies, as well as paths from people to people. The average length of only reaching movies from people l_{pm} can be expressed as:

$$l_{pm} = \frac{(l_r(N_P(N_P - 1) + N_P N_M) - l_{pp} N_P (N_P - 1))}{N_P N_M} \tag{3}$$

3.7 Caveats with the Newman-Strogatz-Watts Equations

There are various problems with using the above formulas in a realistic setting [21]. First, unlike most results in random graph theory, the formulas do not include any guarantees and/or confidence levels. Second, all the equations above are obtained over the ensemble of random graphs that have the given degree distribution, and hence assume that all such graphs are equally likely. The specificity of the jumping connections construction implies that the G_s and G_r graphs are poor candidates to serve as a typical random instance of a graph.

In addition, the equations utilizing N_P and N_M assume that all vertices are reachable from any starting vertex (i.e., the graph is one giant component). This will not be satisfied for very strict jumping constraints. In such cases, Newman, Strogatz, and Watts suggest the substitution of these values with measurements taken from the largest component of the graph. Expressing the size of the components of the graph using generating functions is also suggested [36]. However, the complexity of jumps such as the hammock can make estimation of the cluster sizes extremely difficult, if not impossible (in the Newman-Strogatz-Watts model). We leave this issue to future research.

Finally, the Newman-Strogatz-Watts model is fundamentally more complicated than traditional models of random graphs. It has a potentially infinite set of parameters (p_k), doesn't address the possibility of multiple edges, loops and, by not fixing the size of the graph, assumes that the same degree distribution sequence applies for all graphs, of all sizes. These observations hint that we cannot hope for more than a qualitative indication of the dependence of the average path length on the jump constraints. In the next section, we describe how well these formulas perform on two real-world datasets.

4 Experimental Results

We devote this section to an investigation of two actual datasets from the movies domain; namely the EachMovie dataset, collected by the Digital Equipment Corporation (DEC) Systems Research Center, and the MovieLens [43] dataset developed at the University of Minnesota. Both these datasets were collected by asking people to log on to a website and rate movies. The time spent rating movies was repaid by providing predictions of ratings for other movies not yet seen, which the recommendation engines calculated based on the submitted ratings and some other statistical information.

The datasets contain some basic demographic information about the people (age, gender, etc) as well the movies (title, genre, release date, etc). Associated with each person and movie are unique ids. The rating information (on a predefined numeric scale) is provided as a set of $3-$tuples: the person id, the movie id, and the rating given for that movie by that person. Some statistics for both the datasets are provided in Table 1. Notice that only a small

Table 1: Some statistics for the EachMovie and MovieLens datasets.

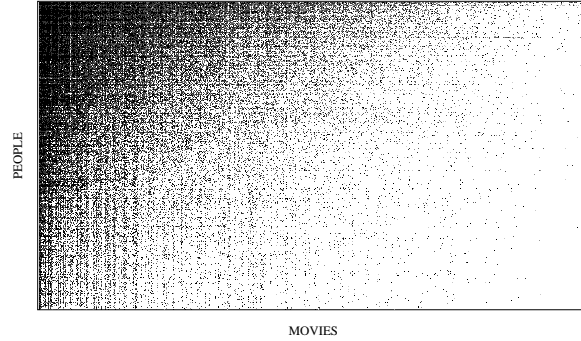| Dataset	Number of people	Number of movies	Sparsity	Connected?
MovieLens	943	1,682	93.70%	Yes
EachMovie	61,265	1,623	97.63%	Yes

Figure 6: Hits-buffs structure of the (reordered) MovieLens dataset.

number of actual ratings are available (as a fraction of all possible combinations), and yet the bipartite graphs of people versus movies are connected, in both cases.
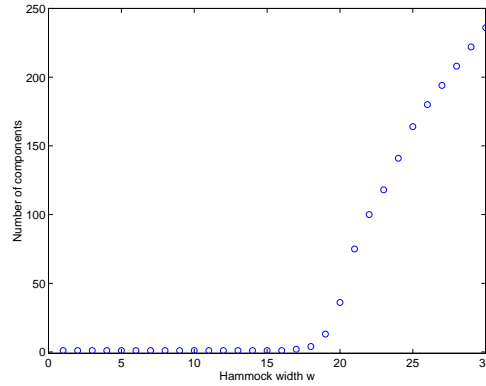
4.1   Preliminary Investigation

Both the EachMovie and MovieLens datasets exhibit a *hits-buffs* structure. Assume that people are ordered according to a *buff index b*: A person with buff index 1 has seen the most number of movies, a person with buff index 2 has seen the second most number of movies, and so on. For example, in the EachMovie dataset, the person with buff index 1 has seen 1,455 movies from the total of 1,623. These 1,455 movies have, in turn, been seen by 61,249 other people. Thus, within two steps, a total of 62,705 nodes in the graph can be visited; with other choices of the intermediate buff node, the entire graph can be shown to be connected in, at the most, two steps. The MovieLens dataset satisfies a similar property.

Furthermore, the relationship between the buff index b and the number of movies seen by the buff $P(b)$ follows a power-law distribution, with an exponential cutoff:

$$P(b) \propto b^{-\alpha} e^{-\frac{b}{\tau}}$$

For the EachMovie dataset, $\alpha \approx 1.3$ and $\tau \approx 10,000$. Similar trends can be observed for the hits and for the MovieLens dataset. Graphs with such power-law regimes can thus form small-worlds [4] as evidenced by the short length between any two people in both MovieLens and EachMovie. To better demonstrate the structure, we reorder the people and movie ids, so that the relative positioning of the ids denotes the extent of a person being a buff, or a movie being a hit. For example, person id 1 refers to the person with buff index 1 and movie id 1 refers to the movie with hit index 1. Figure 6 illustrates the hits-buffs structure of the MovieLens dataset.

Figure 7: Effect of the hammock width on the number of components in the G_r graph induced from the MovieLens dataset.

4.2 Experiments

The goal of our experiments is to investigate the effect of the hammock width w on the average characteristic path lengths of the induced G_s social network graph and G_r recommender graph for the above datasets. We use versions of the EachMovie and MovieLens datasets sanitized by removing the rating information. So, even though rating information can easily be used by an appropriate jump function (an example is given in [2]), we explore a purely connection-oriented jump in this study. Then, for various values of the hammock width w, we form the social network and recommender graphs and calculate the degree distributions (for the largest connected component). This was used to obtain the lengths predicted by equations 1 and 2 from Section 3. We also compute the average path length for the largest connected component of both the secondary graphs using parallel implementations of Djikstra's and Floyd's algorithms. The experimental observations are compared with the formula predictions.

4.2.1 MovieLens

Fig. 7 describes the number of connected components in G_r as a result of imposing increasingly strict hammock jump constraints. Up to about $w = 17$, the graph remains in one piece and rapidly disintegrates after this threshold. The value of this transition threshold is not surprising, since the designers of MovieLens insisted that every participant rate at least $\kappa = 20$ movies. As observed from our experiment results, after the threshold and up to $w = 28$, there is still only one giant component with isolated people nodes (Fig. 8, left). Specifically, the degree distributions of the MovieLens social network graphs for $w > 17$ show us that the people nodes that are not part of the giant component do not form any other connected components and are isolated. We say that a jump *shatters* a set of nodes if the vertices that are not part of the giant component do not have any edges. This aspect of the formation of a giant component is well known from random graph theory [9]. Since our construction views the movies as a secondary mode, we can ensure that only the strictest hammock jumps shatter the N_M movie nodes. Fig. 8 (right) demonstrates that the movie nodes are not stranded as a result of hammock constraints up to hammock width $w = 29$.

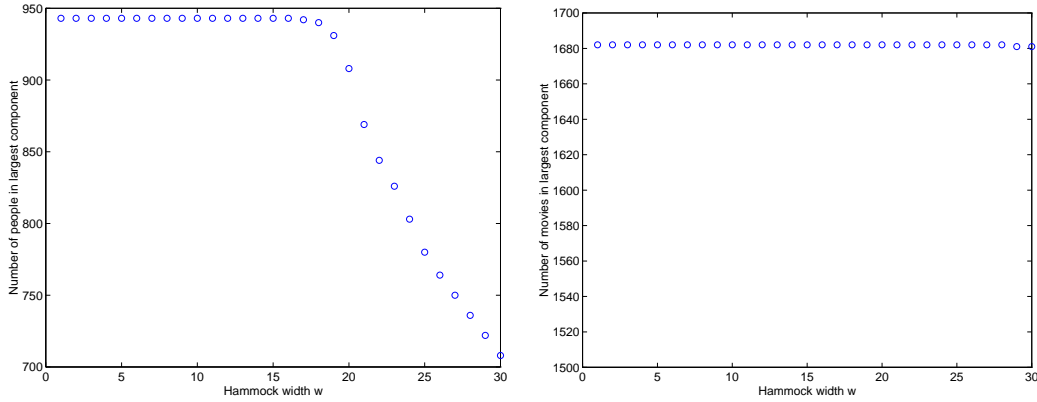The comparison of experimental observations with formula predictions for the l_{pp} and l_r

Figure 8: (left) Effect of the hammock width on the number of people in the largest components in the MovieLens G_r graph. (right) Effect of the hammock width on the number of movies in the largest components in the MovieLens G_r graph.
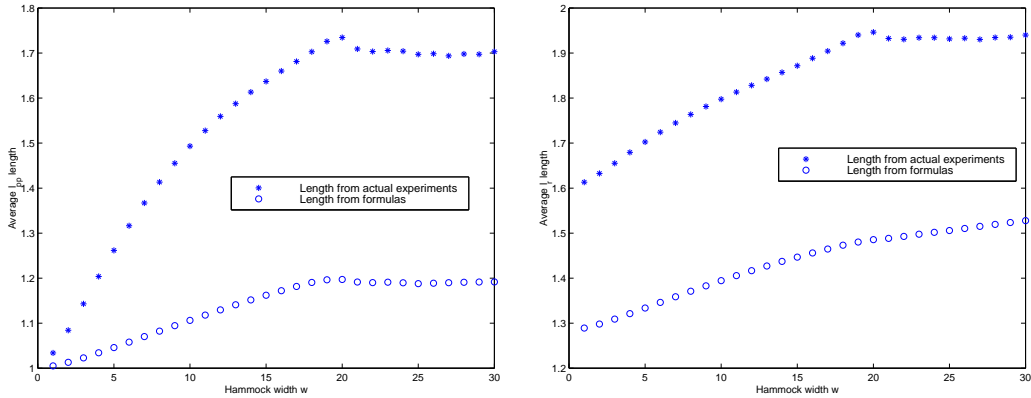

Figure 9: (left) Comparison of the l_{pp} measure (MovieLens) from actual computations and from the formulas. (right) Comparison of the l_r measure (MovieLens) from actual computations and from the formulas.

lengths are shown in Fig. 9. The graphs for l_{pp} share certain important characteristics. The increase in length up to the threshold point is explained by the fact that edges are removed from the giant component, meaning paths of greater length have to be traversed to reach other nodes. After the threshold, the relative stability of the length indicates that the only edges lost are those that are associated with the stranded people nodes. We attribute the fact that the lengths are between 1 and 2 to the hits-buffs structure, which allows short paths to almost any movie. Notice that while the formulas capture the qualitative behavior of the effect of the hammock width, it is obvious from Fig. 9 (left) that they postulate significantly less clustering than is actually observed. A possible explanation is given later in this section.

The comparison of the experimental observations and formula predictions for l_r (Fig. 9, right) show substantially better agreement, as well as tracking of the qualitative change. Once again, the formulas assume significantly less clustering than the actual data. In other words, the higher values of lengths from the actual measurements indicate that there is some source of clustering that is not captured by the degree distribution, and so is not included
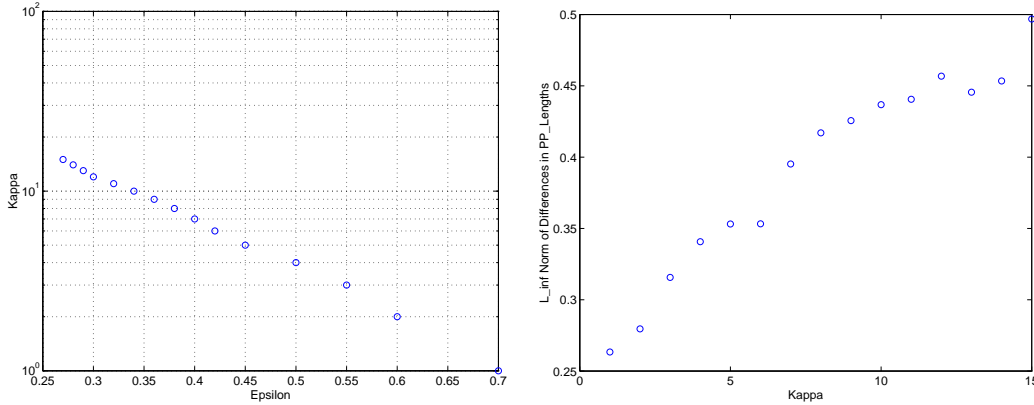
Figure 10: (left) Calibrating the value of ϵ in the synthetic model of EachMovie produces datasets with required specifications on minimum rating κ. (right) Comparison of the l_{pp} measure (EachMovie) from actual computations and from the formulas for varying values of κ.

in the formulas.

4.2.2   EachMovie

The evaluation of the EachMovie data is more difficult owing to inconsistencies in data collection. For example, the dataset was collected in two phases, with entirely different rating instructions and scales in the two situations, and also contains duplicate ratings. We concentrate on the portion of the data collected in 1997 and use a synthetic dataset that has the same sparsity and exponent of the power-law as this reduced dataset. Specifically, our dataset includes 500 people and 75 movies and has the property that the person with buff index b has seen the first $\lceil 75b^{-\epsilon} \rceil$ movies (recall that the movies are also ordered according to their hit id). An $\epsilon = 0.7$ produces a dataset with a minimum rating of 1 movie (95.5% sparse), while $\epsilon = 0.27$ produces a minimum rating of 15 movies (with sparsity 76.63%). The choice of ϵ thus provides a systematic way to analyze the effect of the minimum rating constraint κ (see Fig. 10, left). In addition, for each (person, movie) edge of these synthetic graphs, we generate a uniform (discrete) random variate in $[0, 10]$ and rewire the movie endpoint of the edge if this variate is < 2. This device models deviations from a strict hits-buffs distribution. We generated 15 such graphs and ensured that they were connected (in some cases, manual changes were made to ensure that the graph was connected). These graphs served as the starting points for our analysis. For each of these 15 graphs, we vary the hammock width w from 1 to 25 and repeat the length calculations (using both experiments and formula predictions) for the social network and recommender graphs.

Like the MovieLens experiment, the formulas for EachMovie predict shorter l_{pp} lengths (and consequently, lesser clustering) than observed from actual experiments. To characterize the mismatch, for each of the 15 values of κ, we express the differences between the formula predictions and experimental observations as a vector (of length 25, for each of the 25 values of hammock width w). The L_∞ norm of this vector is plotted against κ in Fig. 10 (right). Notice the relatively linear growth of the discrepancy as κ increases. In the range of κ considered, the hammock constraints shatter the graph into many small components, so
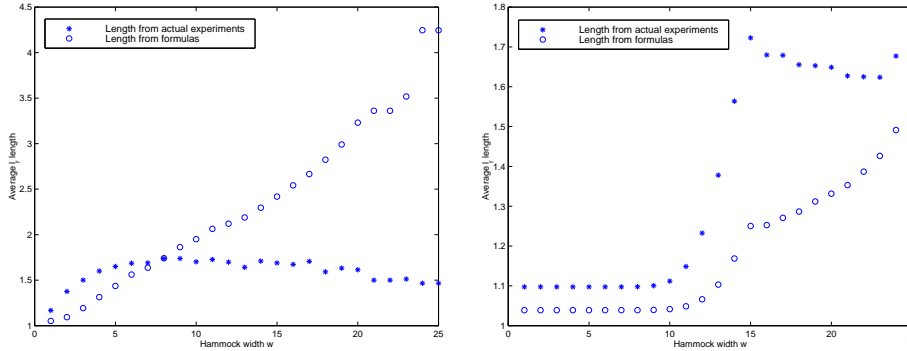
Figure 11: Comparison of the l_r measure (EachMovie) from actual computations and from the formulas for minimum rating constraint $\kappa = 1$ (left) and minimum rating constraint $\kappa = 15$ (right).

the formulas are applied over ever-decreasing values of n, rendering them ineffective. For example, for $\kappa = 15$, a hammock width $w = 25$ shatters the graph into 53 components. Since the hammock width w varies in $[1, 25]$ over a range of $[1, 15]$ for κ, we have reason to believe this graph will taper off when extrapolated to higher values of κ. While this experiment does not provide any new insight, it hints at a fairly bounded growth in the L_∞ discrepancies for l_{pp} lengths.

The comparisons for the lengths in the recommender graph tell a different story (see Fig. 11). At low values of κ, the l_r length calculated from formulas is highly erroneous for even medium values of hammock width w (Fig. 11, left) whereas we see the now familiar effect of assuming too little clustering for high values (≥ 12) of κ (Fig. 11, right). In particular, for $\kappa = 1, w = 25$, the formulas predict an average l_r length of 4.24! This is counter-intuitive given that the largest component for this shattering itself consists of only 4 people nodes (and 71 movie nodes).

This problem arises because the Newman-Strogatz-Watts model does not prohibit a multiplicity of edges between a pair of nodes. Let us look at this situation more closely. For $\kappa = 1, w = 25$, the largest component has the degree distribution given in Table. 2. Notice that the nodes with outdegree 0 are obviously the movie nodes and the nodes with non-zero outdegree are the people nodes. Thus, two of the people are each connected to two people, and two of the people are each connected to three people. A graph that satisfies this property is shown in the left of Fig. 12. The origin of the value of 4.24 can be traced back, rather to the Newman-Strogatz-Watts model's postulation of the unlikely culprit graph shown in the right of Fig. 12. Notice that this graph satisfies the exact same distribution, but by allowing multiple edges many movie nodes are counted more than once, and with greater (ever increasing) path lengths. One cycle between two people can thus effect two extra hops in the length calculations, rendering the estimates inaccurate. As observed from our results, as κ increases, the largest component increases in size. For example, when $\kappa = 15$ and $w = 25$, the largest component has 53 people whereas for $\kappa = 1$ and $w = 25$, there are only 4 people in the largest component. With increase in largest component size for higher values of κ, the proportion of such pathological graphs decreases; hence the observed (qualitative) agreement between actual and predicted values.

Table 2: Joint degree distribution for the largest component of the recommender graph (EachMovie) when $\kappa = 1, w = 25$. Only the non-zero entries are shown.

indegree j	outdegree k	p_{jk}
1	0	23/75
2	0	16/75
3	0	13/75
4	0	19/75
2	31	1/75
2	65	1/75
3	37	1/75
3	47	1/75

Figure 12: Two graphs that satisfy the degree distribution given in Table. 2. For simplicity, only the people nodes are shown.

4.3   Discussion of Results

We can make several preliminary observations from the results so far:

1. As Newman, Strogatz, and Watts point out [36], the random graph model defined by degree distributions makes strong qualitative predictions of actual lengths, using only local information about the number of first and second nearest neighbors (z_1 and z_2 from Section 3). We have demonstrated that this holds true even for graphs induced by hammock jumps.

2. For sufficiently large values of κ, the relationship between hammock width w and average l_{pp} length follows two distinct phases: (i) in the first regime, there is a steady increase of l_{pp} up to a threshold $< \kappa$, where only edges are lost; (ii) in the second phase, nodes are shattered but without much effect on the average l_{pp} values. This two-phase phenomenon can thus serve as a crucial calibration mechanism for connecting the number of people to be brought together by a recommender system and the average l_{pp} length. For example, one can define a parametric study such as discussed here for a new domain and proceed to first demarcate the phases. Choices of hammock width w can then be made, depending on the feasibility of realizing an appropriate κ (in real-life) and any desired constraints on l_{pp}.

3. Visualizing the overlap in the social network graphs, as hammock width constraints are increased, leads to interesting observations. As can be seen in the Venn diagram of Fig. 13 (left), the composition of the nodes in G_s appears to be fairly robust. Smaller and smaller graphs are created, but with a common core, as illustrated by the 'clam shells' in the diagram. In other words, there is not a handful of people or movies who
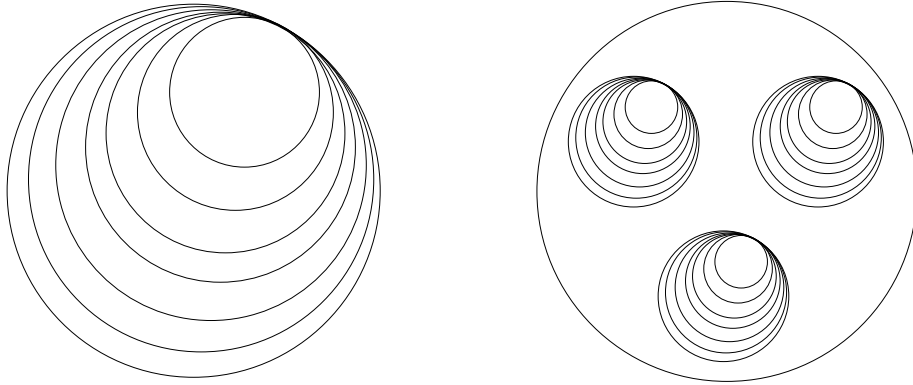
Figure 13: (left) 'Clam shells' view of the social networks G_s induced by various hammock widths. Increasing hammock width leads to smaller networks. (right) a hypothetical situation likely to be caused by the presence of a few strong points in the graph.

form a cutset for the graph — there is no small group of people or movies whose removal (by a sufficiently strict hammock width) causes the entire graph to be broken into two or more pieces (see right of Fig. 13). This confirms observations made elsewhere [6] that power-laws are a major factor ensuring the robustness and scaling properties of graph-based networks. Our study is the first to investigate this property for graphs induced by hammock jumps. However, we believe this robustness is due to the homogeneous nature of ratings in the movie domain. In other domains such as music CDs, where people can be partitioned by preference, the diagram on the right of Fig. 13 would be more common.

4. The average l_r lengths are within the range $[1, 2]$ in both formula predictions and experimental results. Caution has to be exercised whenever the graph size n gets small, as Fig. 12 shows. In our case, this happens when the hammock width constraint w rises above the minimum rating constraint κ. Of course, this behavior should also be observed for other strict jumps.

5. Both of the l_{pp} and l_r formulas postulate consistently less clustering than observed in the real data. We attempt to address this below. The typical random graph model has a Poisson distribution of edges [9], whereas, as seen earlier, real datasets exhibit a power-law distribution [13, 16]. The power-law feature is sometimes described as 'scale-free' or 'scale-invariant' since a single parameter (the exponent of the law) captures the size of the system at all stages in its cycle (the x-axis of the law). A log-log plot of values would thus produce a straight line, an effect not achievable by traditional random graph models. Barabási and Albert [6] provide two sufficient conditions for this property: growth and preferential attachment. *Growth* refers to the ability of the system to dynamically add nodes; random graph models that fix the number of nodes are unable to expand. *Preferential attachment* refers to the phenomenon that nodes that have high degrees have a greater propensity of being linked to by new nodes. In our case, a movie that is adjudged well by most people is likely to become a hit when additional people are introduced. Barabási and Albert refer to this as a "rich get richer" effect [6].
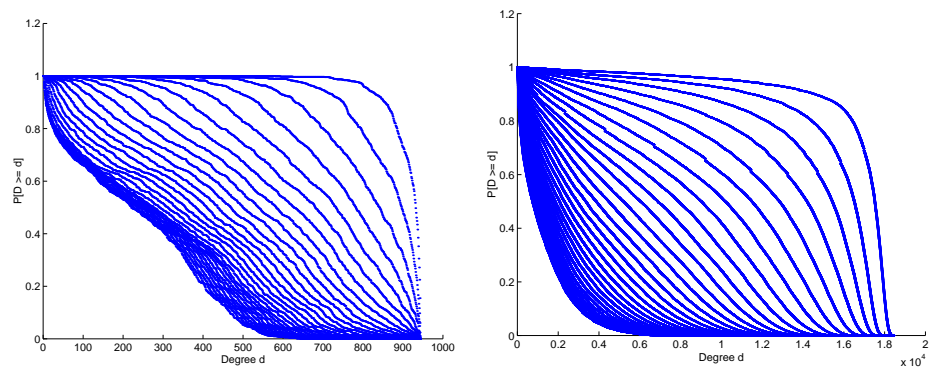
Figure 14: Cumulative frequency distribution (of degrees) in G_s as a function of the degree for (left) MovieLens and (right) EachMovie datasets, for hammock widths from 1 to 30.
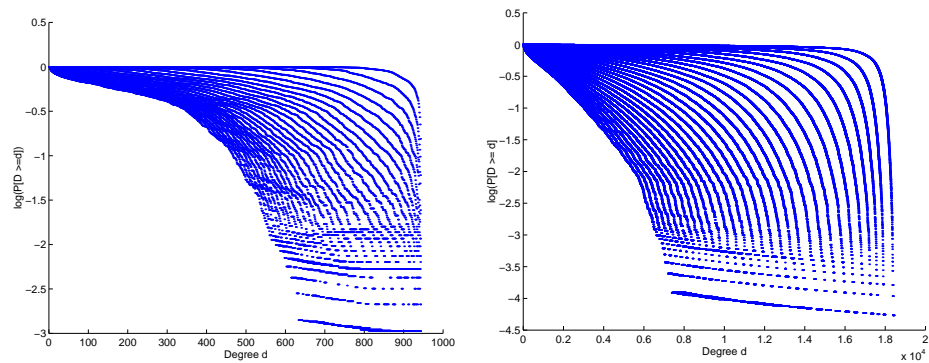

Figure 15: Logarithm of the cumulative frequency distribution (of degrees) in G_s as a function of the degree for (left) MovieLens and (right) EachMovie datasets, for hammock widths from 1 to 30.

To characterize the possible causes in our domain, we consider the distribution of degrees in the social network graphs G_s of both MovieLens and (the actual) EachMovie datasets (see Fig. 14). In the figure, lines from top to bottom indicate increasing hammock widths. Both datasets do not follow a strict power law for the entire range of the hammock width w. For low values of w, there is a small and steadily increasing power-law regime, followed by an exponential cutoff. For higher values of w, the graph 'falls' progressively earlier and the CDF resembles a Gaussian or exponential decay, with no power-law behavior. This is evident if the logarithm of the CDF is plotted, as in Fig. 15 where top to bottom indicates increasing hammock widths. Notice the significant cusp in the left side of both graphs, depicting the qualitative change in behavior. These two extremes of deviations from power-law behavior (at low and high values of w) are referred to as *broad-scale* and *single-scale* [4], respectively, to distinguish them from the scale-free behavior of power-laws.

Causes for such deviations from a pure power-law are also understood, e.g. aging and capacity. *Aging* refers to the fact that after a certain point in time, nodes stop accumulating edges. *Capacity* refers to resource-bounded environments where cost and economics prevent the hits from becoming arbitrarily greater hits. These observations have been verified for environments such as airports (capacity limitations on runways), and natural networks (aging caused by people dying) [4].

In our domain, recall that the G_s graphs model the connectivities among people, and are hence indirect observations from an underlying bipartite graph. One possible explanation for the deviations of the connectivities in G_s from from power-law behavior is suggested in a study done by Robalino and Gibney [45], which models the impact of movie demand on the social network underlying a word-of-mouth recommendation. The two new factors introduced there are *expectation* and *homogeneity* (of the social network). The authors suggest that "some movies might end up having low demand, depending on the initial agents expectations and their propagation through the social network" [45]. Furthermore, they assume that negative information (ratings) obtained early in a movie's lifetime can have substantial effect in a homogeneous network (one where individuals trust each others opinions strongly than in other networks). At this point, we are unable to accept or reject this observation due to lack of information about the social dynamics in which the data was collected in MovieLens and Each-Movie.

However, such an effect might not still model the shift in the emphasis from a broad-scale behavior to a single-scale behavior as w increases. Fortunately, this is easy to explain algorithmically from our construction. For higher values of w, the degree distributions resemble more and more a typical random graph (for smaller values of n), which has a connectivity characterized by a fast decaying tail (such as a Poisson distribution). Insisting on greater values of w leads to higher and higher decays, such that for sufficiently large w (relative to κ), no power-law regime is visible [4].

5   Concluding Remarks

This research makes two key contributions. First, we have shown how algorithms for recommender systems can be studied in terms of the connections they make in a bipartite graph.

This view enables a new methodology to conduct experimental analysis and comparison of algorithms. Using this approach, algorithms can be distinguished by the pairs of nodes that are brought together.

Second, we have described the application of our framework to a particular form of jump — the hammock jump. We have demonstrated a two-phase phenomenon for the induced social network graph that allows us to connect the minimum rating constraint κ, the hammock width w, the size of the largest component, and the average person-person length l_{pp}. In particular, the choice of κ determines the phase transition in varying the hammock width w, with κ being an upper bound on w. Once formalized further (see below), this connection will permit tradeoff analysis between choices for the minimum rating constraint, and the strength of jumps as measured by the hammock width.

5.1   Designing a Recommendation Algorithm

Our goal is to develop the ideas presented here into a complete methodology to guide the design and evaluation of recommendation algorithms. The following scenario illustrates a possible usage context.

Assume that the designer is working with a hammock jump on a power-law dataset such as the ones studied here. There are several parameters that can be used as objectives of interest — κ, w, average l_{pp} length, and the types and numbers of nodes brought together by the jump. The designer first fixes one or more of these variables and tunes the remaining to achieve desired performance criteria. For instance, the designer can fix κ to be the minimum number of artifacts that he would like his customers to rate. He can then determine the w that will achieve a certain threshold in l_{pp}. A graph such as Fig. 9 can be used for this purpose. For instance, the graph might reveal that a setting of $w = 5$ will create an average path length of 1.13 between customers. However, this setting of w will also interfere with the number of people brought together by the jump. Lower values of w are better for length but may not be discriminatory enough, since the jump brings too many people into the connected component. To avoid setting too low a w, the designer can cross-calibrate its value with a plot such as Fig. 8, which qualifies how many people can be reached by the jump. For instance, the designer might choose to increase w at the cost of increased path length, so that stronger connections are induced in the social network graph. The designer can then repeat such an analysis with a different degree distribution, if the problem domain so demands.

Note that our work emphasizes connections, or whether a recommendation is possible. This is complementary to considering predictive accuracy, which must be assessed by a field study.

It is pertinent to note that graph analysis helps support the situations described in Section 1. *Scenario 1* involves finding clusters of people with related interests. This can be done by calibrating the hammock width using a plot of the type shown in Fig. 8 (left). *Scenario 2* involves exploring the connectivity properties, perhaps via visualization, of the social network graphs induced by the jumps under consideration (see Fig. 13). *Scenario 3* involves a calibration of κ based on synthetic (or random) graphs generated by the expected rating patterns. Current work is addressing random graph models that would be applicable in this context — see discussion of future work below.
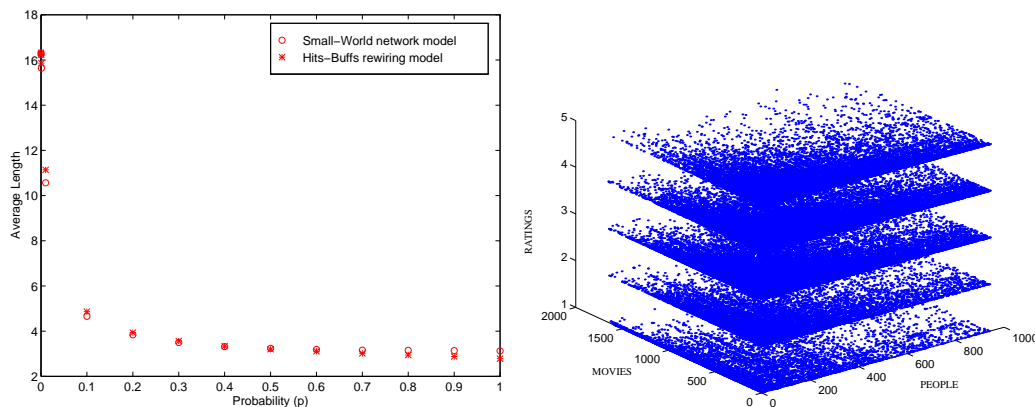
Figure 16: (left) Effect of two rewiring models on the characteristic path length, starting from a regular wreath network. (right) A stratified view of the MovieLens dataset demonstrates hits-buffs structures at all rating levels.

5.2   Future Work

We now describe various opportunities for future research involving graph analysis, more expressive forms of jumps, and development of new random graph models.

Role of Hits-Buffs

The existence of ratings structures such as the hits-buffs distribution and our ability to exploit them (to minimize factors such as the average length) is very crucial to the success of recommender systems. An interesting question is how much of the hits-buffs structure should be present in a dataset to provide high quality recommendations? The answer to this question has implications for current methodologies of data acquisition. Typically, movies (or artifacts) are partitioned into two sets: a *hot set*, that almost everybody is required to rate (to increase commonality of ratings) [2, 19] and a *cold set* that is used to ensure adequate coverage [2]. A more detailed understanding of the hits-buffs structure would provide new methodologies to partition the set of artifacts.

To study this question, we conducted a Watts-Strogatz analysis with a rewiring model that preferentially rewires to some nodes with greater probability. Thus, nodes that are hits or buffs have a greater propensity of being linked to. Fig. 16 (left) shows the results for the average length in a one-mode graph. For small values of rewiring probability p, the results from a preferential rewiring are virtually indistinguishable from a random rewiring, so both of them serve to bring the length down to nearly identical limits. For large values of p, the influence of the hits-buffs structure is evident in the reduced length. It is thus unclear what a base structure should be to explore the role of hits-buffs in length reduction. Recent research on the modeling of dynamic systems shows that power-laws such as hits-buffs are crucial to ensure robustness [13]; more work needs to be done to delineate connections to data collection and desired metrics in recommender systems.

Expressive Jumps

Our experimentation has concentrated on the hammock jump varied by the hammock width parameter. However, it is also possible to consider other jumps within this framework, and compare them by the connections they make for the same recommender dataset. This comparison can concentrate on the clusters of people (or people and movies) that are brought together (illustrated in Fig. 13). For instance, given two algorithms $\mathcal{A}$ and $\mathcal{B}$ parameterized by α, we would be able to make statements of the form

Algorithm $\mathcal{A}$ with $\alpha = 2$ makes the same connections as algorithm $\mathcal{B}$ with $\alpha \leq 6$.

Results of this form would aid in the selection of recommendation algorithms by the relative strengths of the underlying jump.

This approach would clearly point us in the direction of trying to find algorithms that use the strongest jump possible for all datasets. However, these algorithms may be computationally intensive (for instance, recently proposed algorithms that use mixture models [22] and latent variables [25]), while in a particular dataset the jump is actually equivalent to one performed by a much less expensive algorithm. This point is illustrated by the observation in our experimentation that the path lengths in the recommender graph are between 1 and 2, which suggests that algorithms that attempt to find longer paths will end up being equivalent to algorithms that do not. Therefore, the choice of algorithm is strongly dependent on the dataset, and simpler machinery may be more efficient and cost effective than more complex algorithms that would perform better in other contexts.

The need to identify efficient forms of jumping is more important when additional information, such as rating values, is included in jump calculations. Fig. 16 (right) shows the ratings information present in the MovieLens dataset. As can be seen, the hits-buffs structure is stratified by rating values into multiple layers and can be used advantageously both within a ratings layer and across layers. Algorithms can exploit such prior knowledge of rating patterns to make cheaper jumps.

Finally, we make the observation that almost always, *recommendation algorithms never bring disconnected portions of a graph together*, and when they do, it is in only very pathological cases. This observation, in part, leads us to the belief that all recommender jumps depend in some way on hammock jumps. In some cases this dependency is obvious, but others will require more study.

New Random Graph Models

The formulas presented in this paper for the lengths of the social network graph and of the recommender graph are derived from the parameters of the induced G_s and G_r graphs. One possible direction of work is to cast these variables in terms of parameters of the original bipartite graph (or dataset $\mathcal{R}$). However, the Newman-Strogatz-Watts model is very difficult to analyze for all but the simplest forms of jumps. Recall the Aiello-Chung-Lu model [3] for massive graphs modeled after power-laws. Unlike the Newman-Strogatz-Watts model and like traditional random graph models, this model has only two parameters (the intercept and slope of the power-law plotted on a log-log scale). Estimations of graph properties such as diameter have very recently been initiated [34] for this new model and it appears to be a promising candidate for application to recommender systems. In particular, we could aim for a more accurate modeling of the connection between κ and the hammock width constraint

w at which the graph becomes shattered.

Acknowledgements: The authors express their thanks to L.S. Heath for helpful discussions. Acknowledgements are also due to the Compaq Equipment Corporation, which provided the EachMovie dataset and the University of Minnesota, which provided the MovieLens dataset used in our experiments.

References

[1] L.A. Adamic. "The Small World Web". In *ECDL'99, Proceedings of the Third European Conference on Research and Advanced Technology for Digital Libraries*, pages 443–452. Springer-Verlag, 1999.

[2] C.C. Aggarwal, J.L. Wolf, K.-L. Wu, and P.S. Yu. "Horting Hatches an Egg: A New Graph-Theoretic Approach to Collaborative Filtering". In *KDD'99, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 201–212. ACM Press, 1999.

[3] W. Aiello, F. Chung, and L. Lu. "A Random Graph Model for Massive Graphs". In *STOC'00, Proceedings of the ACM Symposium on Theory of Computing*, pages 171–180. ACM Press, 2000.

[4] L.N. Amaral, A. Scala, M. Bathelemy, and H.E. Stanley. "Classes of Behavior of Small-World Networks". *Proceedings of the National Academy of Science, USA*, Vol. 97:pages 11149–11152, 2000.

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

[6] A.-L. Barabási and R. Albert. "Emergence of Scaling in Random Networks". *Science*, Vol. 286:pages 509–512, October 1999.

[7] C. Basu, H. Hirsh, and W.W. Cohen. "Recommendation as Classification: Using Social and Content-Based Information in Recommendation". In *AAAI'98, Proceedings of the Fifteeth National Conference on Artifical Intelligence*, pages 714–720. AAAI Press, 1998.

[8] D. Billsus and M. Pazzani. "Learning Collaborative Information Filters". In *ICML'98, Proceedings of the Fifteenth International Conference on Machine Learning*, pages 46–53. Morgan Kaufmann, 1998.

[9] B. Bollabás. *"Random Graphs"*. Academic Press, London, 1985.

[10] J.S. Breese, D. Heckerman, and C. Kadie. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". In *UAI'98, Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, pages 43–52. Morgan Kaufmann, 1998.

[11] S. Brin and L. Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine". In *WWW'98, Proceedings of the Seventh International World Wide Web Conference*, pages 107–117. Elsevier Science, 1998.

[12] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. "Graph Structure in the Web". In *WWW'99, Proceedings of the Ninth International World Wide Web Conference*, 1999.

[13] J.M. Carlson and J. Doyle. "Highly Optimized Tolerance: A Mechanism for Power Laws in Designed Systems". *Physical Review*, E60(1412), 1999.

[14] S. Chakrabarti, B.E. Dom, S. Ravi Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Klienberg. "Mining the Web's Link Structure". *IEEE Computer*, Vol. 32(8):pages 60–67, August 1999.

[15] P. Erdös and A. Renýi. "On Random Graphs". *Publicationes Mathematicae*, Vol. 6:pages 290–297, 1959.

[16] M. Faloutsos, P. Faloutsos, and C. Faloutsos. "On Power-Law Relationships of the Internet Topology". In *Proceedings of the ACM SIGCOM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 251–262. ACM Press, 1999.

[17] G.W. Flake, S. Lawrence, and C. Lee Giles. "Efficient Identification of Web Communities". In *KDD'00, Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160. ACM Press, 2000.

[18] D. Gibson, J. Kleinberg, and P. Raghavan. "Clustering Categorical Data: An Approach Based on Dynamical Systems". *VLDB Journal*, Vol. 8(3-4):pages 222–236, 2000.

[19] 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 Berkeley, August 2000.

[20] N. Good, J. Scafer, J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl. "Combining Collaborative Filtering with Personal Agents for Better Recommendations". In *AAAI'99, Proceedings of the Sixteenth National Conference on Artifical Intelligence*, pages 439–446. AAAI Press, 1999.

[21] L.S. Heath. Personal Communication, 2001.

[22] D. Heckerman, D.M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. "Dependency Networks for Inference, Collaborative Filtering, and Data Visualization". *Journal of Machine Learning Research*, Vol. 1:pages 49–75, 2000.

[23] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. "An Algorithmic Framework for Performing Collaborative Filtering". In *SIGIR'99, Proceedings of the Twenty Second Annual ACM Conference on Research and Development in Information Retrieval*, pages 230–237. ACM Press, 1999.

[24] W. Hill, L. Stead, M. Rosenstein, and G. Furnas. "Recommending and Evaluating Choices in a Virtual Community of Use". In *CHI'95, Proceedings of the ACM Conference on Human Factors and Computing Systems*, pages 194–201. ACM Press, 1995.

[25] T. Hofmann and J. Puzicha. "Latent Class Models for Collaborative Filtering". In *IJCAI'99, Proceedings of the 16th International Joint Conference on Artificial intelligence*. IJCAI Press, 1999.

[26] H. Kautz, B. Selman, and M. Shah. "ReferralWeb: Combining Social Networks and Collaborative Filtering". *Communications of the ACM*, Vol. 40(3):pages 63–65, March 1997.

[27] B. Kitts, D. Freed, and M. Vrieze. "Cross-Sell: A Fast Promotion-Tunable Customer-Item Recommendation Method Based on Conditional Independent Probabilities". In *KDD'00, Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pages 437–446. ACM Press, 2000.

[28] J. Kleinberg. "Authoritative Sources in a Hyperlinked Environment". *Journal of the ACM*, Vol. 46(5):pages 604–632, September 1999.

[29] J. Kleinberg. "The Small-World Phenomenon: An Algorithmic Perspective". *Nature*, Vol. 406(6798), August 2000.

[30] A. Kohrs and B. Merialdo. "Clustering for Collaborative Filtering Applications". In *Proceedings of Computational Intelligence for Modelling, Control and Automation*. IOS Press, 1999.

[31] J. Konstan, B. Miller, D. Maltz, J. Herlocker, L. Gordan, and J. Riedl. "GroupLens: Applying Collaborative Filtering to Usenet News". *Communications of the ACM*, Vol. 40(3):pages 77–87, March 1997.

[32] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. "Trawling the Web for Emerging Cyber-Communities". In *WWW'99, Proceedings of the Eighth International World Wide Web Conference*, 1999.

[33] Macromedia LikeMinds. "Macromedia LikeMinds Personalization and Performance". URL: `http://ebusiness.macromedia.com/software/likeminds/whitepapers/lm_white_3rd.pdf`.

[34] L. Lu. "The Diameter of Random Massive Graphs". In *SODA'00, Proceedings of the Twelth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 912–921. ACM/SIAM Press, 2000.

[35] B.J. Mirza. "Jumping Connections: A Graph-Theoretic Model for Recommender Systems". Master's thesis, Virginia Tech, February 2001. 64 pages.

[36] M. Newman, S. Strogatz, and D. Watts. "Random Graphs with Arbitrary Degree Distribution and their Applications". Technical Report 0007042, Santa Fe Institute, 2000.

[37] M. O'Conner and J. Herlocker. "Clustering Items for Collaborative Filtering". In *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, 1999.

[38] D. Payton. "Discovering Collaborators by Analyzing Trails through an Information Space". In *Proceedings of the AAAI Fall Symposium on Artificial Intelligence and Link Analysis*, pages 84–87. AAAI Press, 1998.

[39] D. Pennock, E. Horvitz, S. Lawrence, and C. Giles. "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach". In *UAI'00, Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 473–480. Morgan Kaufmann, 2000.

[40] P. Pirolli, J. Pitkow, and R. Rao. "Silk from a Sow's Ear: Extracting Usable Structures from the Web". In *CHI'96, Proceedings of the ACM Conference on Human Factors and Computing Systems*. ACM Press, 1996.

[41] P. Resnick and H. Varian. "Recommender Systems". *Communications of the ACM*, Vol. 40(3):pages 56–58, March 1997.

[42] L. Richard. "Automatic Information Extraction from Documents: A Tool for Intelligence and Law Enforcement Analysts". In *Proceedings of the AAAI Fall Symposium on Artificial Intelligence and Link Analysis*, pages 63–67. AAAI Press, 1998.

[43] J. Riedl and J. Konstan. "GroupLens Research". URL: `http://www.cs.umn.edu/Research/GroupLens/`.

[44] E. Riloff and L. Hollaar. "Text Databases and Information Retrieval". In A.B. Tucker Jr., editor, *"The Computer Science and Engineering Handbook"*, pages 1125–1141. CRC Press, 1996.

[45] D. Robalino and M. Gibney. "A Model of the Impact on Movie Demand of Social Networks and Word of Mouth Recommendation". URL: `http://www.metaculture.net/Searches/movies`, 1999.

[46] 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*, pages 158–167. ACM Press, 2000.

[47] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. "Application of Dimensionality Reduction in Recommender System – A Case Study". Technical Report CS-TR 00-043, Computer Science and Engineering Dept., University of Minnesota, July 2000.

[48] J. Schafer, J. Konstan, and J. Riedl. "Recommender Systems in E-Commerce". In *Proceedings of the First ACM Conference on Electronic Commerce*, pages 158–166. ACM Press, 1999.

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

[50] U. Shardanand and P. Maes. "Social Information Filtering: Algorithms for Automating 'Word of Mouth'". In *CHI'95, Proceedings of the ACM Conference on Human Factors and Computing Systems*, pages 210–217. ACM Press, 1995.

[51] I. Soboroff and C. Nicholas. "Combining Content and Collaboration in Text Filtering". In *Proceedings of the IJCAI'99 Workshop on Machine Learning in Information Filtering*, pages 86–91, 1999.

[52] D. Swanson and N. Smalheiser. "Link Analysis of MedLine Titles as an Aid to Scientific Discovery". In *Proceedings of the AAAI Fall Symposium on Artificial Intelligence and Link Analysis*, pages 94–97. AAAI Press, 1998.

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

[54] L. Ungar and D. Foster. "Clustering Methods for Collaborative Filtering". In *Proceedings of the AAAI-98 Workshop on Recommender Systems*, pages 112–125. AAAI Press, 1998.

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

[56] D. Watts and S. Strogatz. "Collective Dynamics of 'Small-World' Networks". *Nature*, Vol. 393(6):pages 440–442, June 1998.