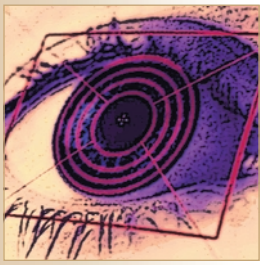

Privacy Risks in Recommender Systems

Recommender system users who rate items across disjoint domains face a privacy risk analogous to the one that occurs with statistical database queries.

Naren Ramakrishnan, Benjamin J. Keller, and Batul J. Mirza
Virginia Tech

Ananth Y. Grama
Purdue University

George Karypis
University of Minnesota

Recommender systems have become important tools in e-commerce. They combine one user's ratings of products or services with ratings from other users to answer queries such as "Would I like X?" with predictions and suggestions. Users thus receive anonymous recommendations from people with similar tastes. While this process seems innocuous, it aggregates user preferences in ways analogous to statistical database queries, which can be exploited to identify information about a particular user. This is especially true for users with eclectic tastes who rate products across different types or domains in the systems.

These *straddlers* highlight the conflict between personalization and privacy in recommender systems. While straddlers enable serendipitous recommendations, information about their existence could be used in conjunction with other data sources to uncover identities and reveal personal details. We use a graph-theoretic model to study the benefit from and risk to straddlers.

Recommender Systems

One common form of recommendation uses a nearest-neighbor algorithm to find correlations between users' preferences.[1] Table 1 shows an example system in which four users each rated a subset of 10 movies. While recommendation algorithms work in various ways, we employ a simple formulation here for illustration: We assume that one person can recommend for another if their ratings agree for a simple majority of commonly rated products. Because Abby and Charles agree on five of seven ratings, we can predict preferences for movies that only one has seen. Because Charles liked *Il Postino*, for instance, we can predict that Abby would as well.

 1089-7801/01/$10.00 ©2001 IEEE

Abby and Charles do not agree sufficiently with Bernie or Daphne to exchange recommendations with them, although some algorithms allow anticorrelations[2] that permit inverse recommendations such as "Abby would like *Il Postino* because Bernie did not."

The nearest neighbor algorithm implies a connection between Abby and Charles as illustrated in Figure 1. This diagram shows a bipartite graph where the vertices are people and movies, and the edges indicate that the person has rated the movie. The common ratings provide the basis for connecting Abby to movies that Charles has rated but she has not.

Note that knowledge of Abby's ratings, the algorithm, and the query results tell us that at least one other person has rated enough items in common with Abby to provide a positive recommendation.

Table 1. Example movie ratings.

Movie	Abby	Bernie	Charles	Daphne
Austin Powers	✗	✓	✓	✓
Braveheart	✓	✗		✗
Castaway	✓	✗	✓	✗
Don Juan DeMarco	✓		✓	✗
Emma	✓	✗	✓	✗
Faceoff	✗	✓	✗	
Goodfellas	✗	✓		
Heathers	✗		✓	
Il Postino		✗	✓	
Jane Eyre	✓	✗	✓	✗

Key: ✓ = Like; ✗ = Dislike

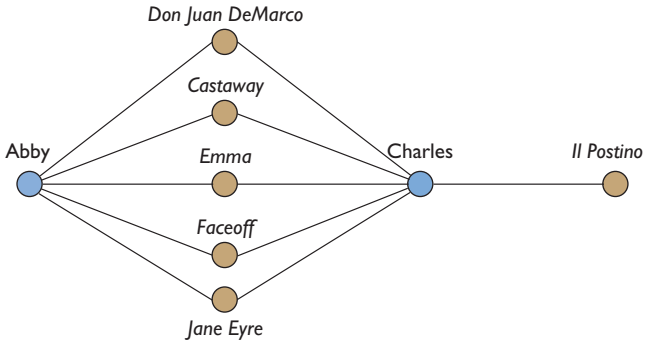

Figure 1. Recommendation relationship. The agreement between ratings from Abby and Charles allows the system to recommend a movie that Abby has not yet rated.

Serendipitous Recommendation

Recommendation algorithms are designed to balance several considerations, including statistical significance, potential for additional revenues and boosting sales, and ability to hold the user's attention. Recommender systems also try to make good use of *serendipity* — the ability to recommend something unexpected but desirable — because in addition to reflecting consumers' buying patterns, the goal is to enable cross-selling or to cater to populations with novel tastes.

An application domain such as books, in which people exhibit stronger preferences and tastes, illustrates this idea clearer. Suppose that Abby reads networking books for work and is also an Indian classical music enthusiast. She would be surprised (and possibly unnerved) to get a recommendation for a book called *Evolution of Indian Classical Music* if all her prior ratings were of networking books, although it would be desirable from the viewpoint of serendipity. Figure 2 illustrates this serendipitous recommendation: the two people who rated (at least) both the networking books that Abby rated also rated the one Indian classical music title.

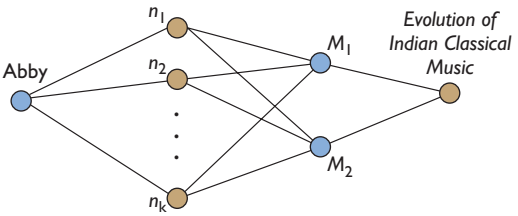

Figure 2. Serendipitous recommendation. Abby rates k networking books and receives an unexpected recommendation for a book on Indian classical music (through commonality with two people who have rated the music text).

Of course, many other people might have rated the classical music book and the networking books, but the results of this query could allow Abby to discover the people who bridge these disparate topics. By masquerading as another user, she could add ratings incrementally to determine the smallest set of ratings necessary to generate this recommendation. Note that this probing would not perturb the original setting because she need only rate networking books from the new personas. This is because Abby's probing does not create new connections, but allows her to make observations about the aggregate ratings of other people. In fact, with sufficient knowledge of the algorithm, Abby could actually estimate the (aggregate) rating of the

Table 2. Customer history database.

Cust ID	Gender	OS	NW	CM	MT	Age	$Sales	Profession	Credit card
1	M	✓	✓			21	85	Student	...
2	F			✓		23	90	Engineer	...
3	M	✓	✓			32	65	Programmer	...
4	M		✓	✓		31	124	Professor	...
5	M			✓	✓	26	65	Student	...
6	F			✓	✓	24	55	Student	...

Key: OS = operating systems book, NW = networking book, CM = Indian classical music book, MT = music theory book

other people (using her observations and the agreement formulas employed by the algorithm).

Even with more variety in the ratings, this type of probing can't determine whether a single straddler has bridged the domains, although the likelihood of many straddlers decreases with additional information. Suppose, for instance, that the system recommended several networking books, the Indian classical music text, and several books on French cuisine in response to the original query. Abby could probe to determine the subset of ratings for networking books that eliminates or adds the French cuisine titles but maintains the music text. Her knowledge of this rating information poses no risk by itself, but used in conjunction with other information, it fills a gap in knowledge that could be key to identifying straddlers in the system. This is similar to the privacy risk inherent in allowing statistical database queries where it is possible to make inferences from combinations of queries.

Identity Compromise

The risk is compounded somewhat if Abby is also a consultant with access to the database of information about people and purchases. This is a realistic possibility given that e-commerce sites periodically provide databases to third-party consultants for data mining, intrusion detection, and statistical reporting. In some cases, personalization services are provided entirely by external firms. The need to balance legitimate use and potential for malicious use is well recognized in the database literature.[3] Indeed, consultants like Abby are traditionally allowed to enquire about records by no other means than statistical queries (using AVERAGE, COUNT, and SUM operations, for example). Typical analysis with a database such as the one in Table 2 involves determining the total sales for one title or the demographics of those interested in books of a given type. Queries such as "what is the average age of people who buy the networking book (NW)?" or "how much do music theory (MT) people spend at this site?" are legitimate queries whose answers will help the site better serve its customers.

By issuing a COUNT query, Abby could verify that there is only one person who has rated both the networking and classical music books. She could then query the database for the average age of people who have bought both books, which would reveal the age of customer 4 and compromise a personal detail! Attributes stored in other data fields could be similarly identified.

Level of risk. The reader might argue that recommender systems do not add new privacy concerns if Abby already has access to the database. She would ordinarily be hard pressed, however, to identify an eclectic group such as "people who have rated both NW and CM" because she can only issue statistical queries. Recommender systems, on the other hand, confirm such people's existence and help Abby to focus her snooping. In some application domains, such as committee memberships or voting records, simply being able to deduce a connection can constitute a breach of privacy.

The nature of the attack we have described has two components:

- using explanations of recommendations to deduce connections, and
- combining connection information with other data to reveal people's personal details.

The second component is well studied in the database literature as the problem of *inference control*,[3] which seeks to limit the inferences a malicious hacker can make by querying a statistical database. This usually assumes that the hacker has already deduced some connection that uniquely identifies a particular individual.

Explanations in recommender systems provide hackers a key source of information for posing such malicious queries. This is especially true when the explanations provide count or descriptive informa-

tion that indicates the recommendation's strength based on the number of people involved. The explanations can thus reveal information analogous to that provided by statistical database queries. Combined with other knowledge, this information could be used to identify a specific person.

Greatest susceptibility. Clearly, most explanations raise no privacy concerns. Only those that reveal connections between disparate groups of people present a true risk, and even this is drastically diminished when a large number of such connections exist (if several people rated networking books and books on Indian classical music, for example). The straddler whose ratings first create this connection faces the greatest risk.

It is commonly believed that we can secure a database by requiring that all queries posed by Abby involve at least, say, 10 people, or by controlling the overlap between successive queries posed by her. Unfortunately, this is not enough. Denning et al.[3] describe a relatively simple mechanical procedure for constructing a "tracker" that can provide statistics for any arbitrary query set, especially when individuals are uniquely characterized by conditions on attributes in a database schema. Abby can thus construct a tracker to determine "the average age of people who rate NW and CM," without asking for it explicitly.

Because of their novel rating patterns, straddlers are most easily characterized and most susceptible to tracking procedures. Trackers can also characterize other groups of people, but they can't compromise individual attributes. A tracker for "the average age of people who buy NW," for example, raises no privacy concerns.

Graph-Theoretic Approach

We could determine the benefits from and risks to straddlers by performing a detailed analysis with a particular recommendation algorithm, but it would be difficult to draw general conclusions from this approach. Instead, we aim for an algorithm-independent analysis by employing a graph-theoretic model called *jumping connections*.[4]

General Model

We can view all recommendation algorithms as mechanisms for positing connections between people based on some commonality. This could be based on an overlap of rated artifacts, agreement on actual ratings, or a more sophisticated measure of correlation or statistical similarity.[5] Our model ignores these distinctions by viewing recommendation only as a way to make connections; individual algorithms simply posit different connections.

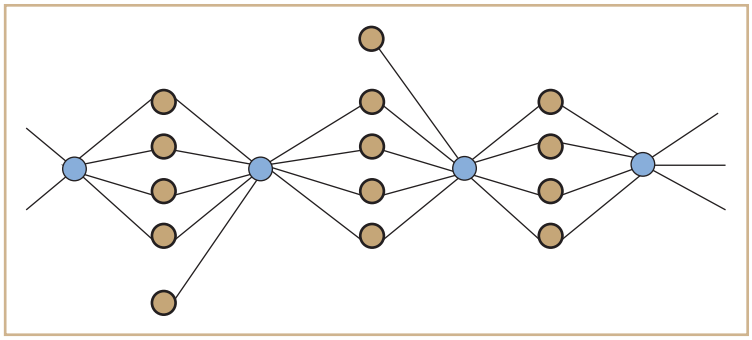

Figure 3. Hammock path. This sequence of hammocks connects people who share at least four ratings.

In the graph representation of ratings shown in Figure 1, the common ratings form what we call a *hammock*. A hammock of width w connects two people if they share at least w ratings. Figure 3 illustrates a hammock path of length l, in which a sequence of l hammocks is employed to connect two people. Different algorithms use hammocks in different ways to make recommendations, but our hypothesis is that hammocks underlie most recommendation approaches.

Nearest-neighbor algorithms, such as those used by GroupLens (http://www.cs.umn.edu/Research/GroupLens/), LikeMinds, and Firefly, employ an implicit hammock path of length 1. Additional constraints are typically imposed regarding agreement in rating values. The horting algorithm uses hammock paths of length greater than 1.[2]

Recommendation is usually studied by comparing predicted ratings from hammock paths versus user feedback, but we qualify recommendations according to hammock width and path length instead. This allows us to make statements of the form, "artifact X can be recommended for person Y when $l = 2$ and $w \leq 5$." Notice, however, that our model does not emphasize how to transform the individual ratings in a hammock into a prediction.

In practice, there are likely to be multiple paths between artifact X and person Y with various constraints on w and l. Intuitively, a wider hammock seems likely to generate better recommendations because we have more common ratings to work with (although we must be careful when considering correlations between ratings[1]). By insisting on a wide hammock, however, we might have to traverse longer paths to reach a particular artifact from a given person.[4] Still, recommendations involving shorter path lengths are preferred over longer paths because they are easier to explain. From a graph-theoretic viewpoint, the parameters

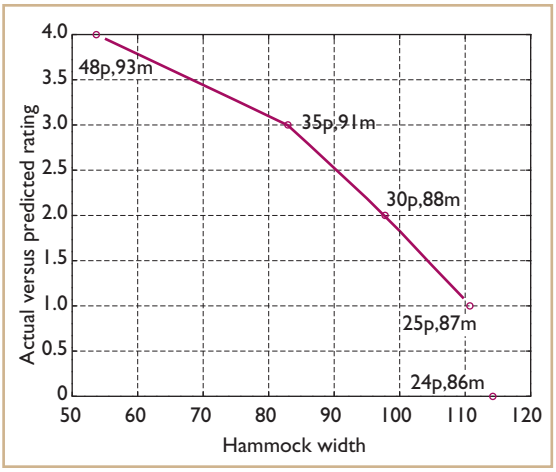

Figure 4. Hammock width influence on the LikeMinds algorithm. The quality of recommendation improves as hammock width is increased, but as the annotations show, the percentage of reachable people (p) and reachable movies (m) also decreases.

w and l determine the reachability of artifacts from each person and indirectly provide a measure of expected prediction quality.

Suitability

To test whether hammocks of greater width and shorter path lengths lead to better recommendations, we analyzed the effects of w and l on prediction quality with the LikeMinds and horting algorithms.[2]

LikeMinds algorithm. Because the LikeMinds algorithm uses a single hammock, we can isolate the role of w in recommendation quality. For this study, we used the public-domain MovieLens dataset (available at http://movielens.umn.edu/), which consists of 943 people, 1682 movies, and at least 20 ratings from every person for movies of their choice. We masked each rating, in turn, and obtained a prediction for that rating based on the remaining data.

To predict the rating of movie X for person Y, LikeMinds computes a metric called the agreement scalar (between Y and every other person who has rated X). The algorithm uses the ratings from the person with the highest agreement scalar to compute the recommendation. We recorded w as the number of common ratings between person Y and the person with the highest agreement scalar. Figure 4 shows a plot of the average discrepancy between the actual ratings and those predicted by Likeminds for each hammock width.[2] The results indicate that, for this algorithm, wider hammocks contribute to lower discrepancies and better ratings. Notice that LikeMinds uses hammocks not just to model commonality, but also to represent agreement between the rating values spanning a hammock. While we can certainly get poor recommendations even with a wide hammock (perhaps involving noisy ratings or a faulty aggregation procedure), hammock widths did influence overall prediction quality.

Increasing the hammock width, however, also increases restrictiveness in making connections, which limits connections and renders some people unreachable. Figure 4 shows the percentage of reachable people and movies for various hammock widths. A w of 53, for instance, reaches only 48 percent of the people and 93 percent of the movies. By the time we observe a strong correlation between predicted and actual ratings (after $w \approx 110$), less than 25 percent of the people and only about 86 percent of the movies can be reached. Our model thus captures the tradeoff between smaller w values, which reach more people (and movies), and larger w values, which are more accurate.

Horting algorithm. The horting algorithm uses explicit hammock paths of varying length to provide recommendations. To study the effect of path length l, we fixed the hammock width w at 113 and analyzed paths of varying lengths from people to movies. The algorithm uses a transformation technique similar to one used by LikeMinds to make predictions from others' ratings. For the MovieLens data set, we found that all paths involved 1, 2, or 3 hops between people and a final hop to the recommended movie, resulting in path lengths of 2, 3, and 4 for all recommendations. As shown in Figure 5, greater path lengths caused a faster-than-linear decay in prediction quality for a given w.

Findings. These results support our hypothesis that wider hammocks and shorter paths provide better ratings, and confirm our model's applicability to studying recommendation. Hammock widths are determined by rating patterns that ensure significant overlap of rated artifacts. For the rating pattern in the MovieLens dataset, there is enough overlap to provide a recommendation of any movie for any person as long as $w \leq 17$.[4] Hammock paths are generally shorter when the rating graph includes a large number of connections, and recommendations are almost always possible in the MovieLens dataset using a path of no longer than 3.

These specific results follow from the power-law degree distribution of the MovieLens rating patterns. The rating pattern comes from *preferential attachment*. That is, some movies are rated by almost everyone, and some people rate almost all movies. The implications of the power-law pattern

on w and l are analyzed in detail elsewhere.[4]

Role of Straddlers

Figure 6a (next page) describes the power law behavior in data sets such as MovieLens. The vertices in the middle represent the movies (the four blue ones are the most popular) and the black vertices on the edges represent the people (the three on the left rated every movie). To understand the role of straddlers, we experimented with increasingly greater hammock widths to determine the resulting connections. As mentioned, the system can recommend any movie for any person as long as $w \leq 17$; in other words, the graph induced by hammocks for widths of less than 17 is *connected*.

Figure 6b shows a Venn diagram in which each circle denotes a set of people who remained connected under the hammock condition. As we increased w, some people were stranded and the size of the connected component decreased, resulting in the smaller circles. Straddlers in this scenario showed no strong allegiance to any genre, having rated relatively few movies in each. As they were disconnected from the rest of the network, these straddlers could not be identified through explanations. They faced no privacy risk, but neither could they receive recommendations.

Privacy risks did exist, however, in situations where most people exhibited a preference for a particular kind of movie. Figure 6c illustrates several possibilities for straddlers. The three subgraphs were connected by a relatively small number of ratings because only a few people rated artifacts in subgraphs other than their own. The risk here is that such rating patterns might let us identify a person whose ratings get us from one domain to another. Once again, we verified this by slowly increasing w. As Figure 6d shows, we continued until we broke the connected graph into three disconnected portions, each of which then behaved as a regular power law network. The straddlers were most susceptible to discovery just prior to this decomposition, because they alone kept the graphs connected.

Note that the ratings themselves do not qualify a user as a straddler; we can make such classifications only in relation to other people's ratings. Many people might have rated items across several domains, for example, but perhaps only a few had enough ratings to satisfy the current w. These people might not be perceived as straddlers at a narrower hammock width. The merging of data collected from different settings, such as the recent purchase of eToys consumer data by another retail giant, might also create a graph with straddler nodes that could be used to identify personal data about the customers.

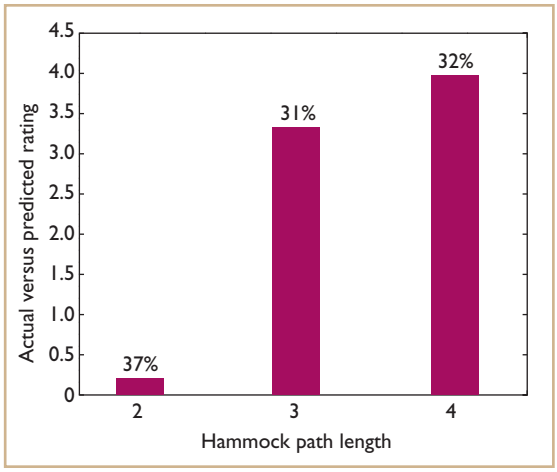

Figure 5. Hammock path length influence on the horting algorithm. As path length increases, the quality of recommendation decreases. The annotations denote the number of recommendations possible for each level of length.

Recommender systems validate that there is safety in numbers, or at least in homogeneous tastes (as in power law graphs). The likelihood of identifying any one user decreases as more people rate the same things. The risk remains if the w constraint is used to weed some people out, but it is less likely that additional information could be used to identify a single person in this case (see Figure 6b).

Benefit-Risk Analysis

We have shown that a user benefits most from recommendations based on wide hammocks and short path lengths. Of course, that requires the user to provide a larger number of ratings, which increases the risk of becoming a straddler. We must therefore relate the number of submitted ratings to the benefit and risk inherent to recommendation.

Modeling Benefit

A formula for determining the benefit of a given recommendation path should capture our preference for wider hammocks and shorter path lengths. The contours of the graphs in Figures 4 and 5 suggest that benefit is best modeled by a linear dependence on w and a nonlinear, inverse dependence on l. In addition, research in diffusion processes[6] and social networks[7] supports the theory of nonlinear dependence of interaction quality on length. We thus define the benefit of a recommendation path as:

$$\text{benefit} = \frac{w}{l^2}$$

Note that this formula gives more weight to

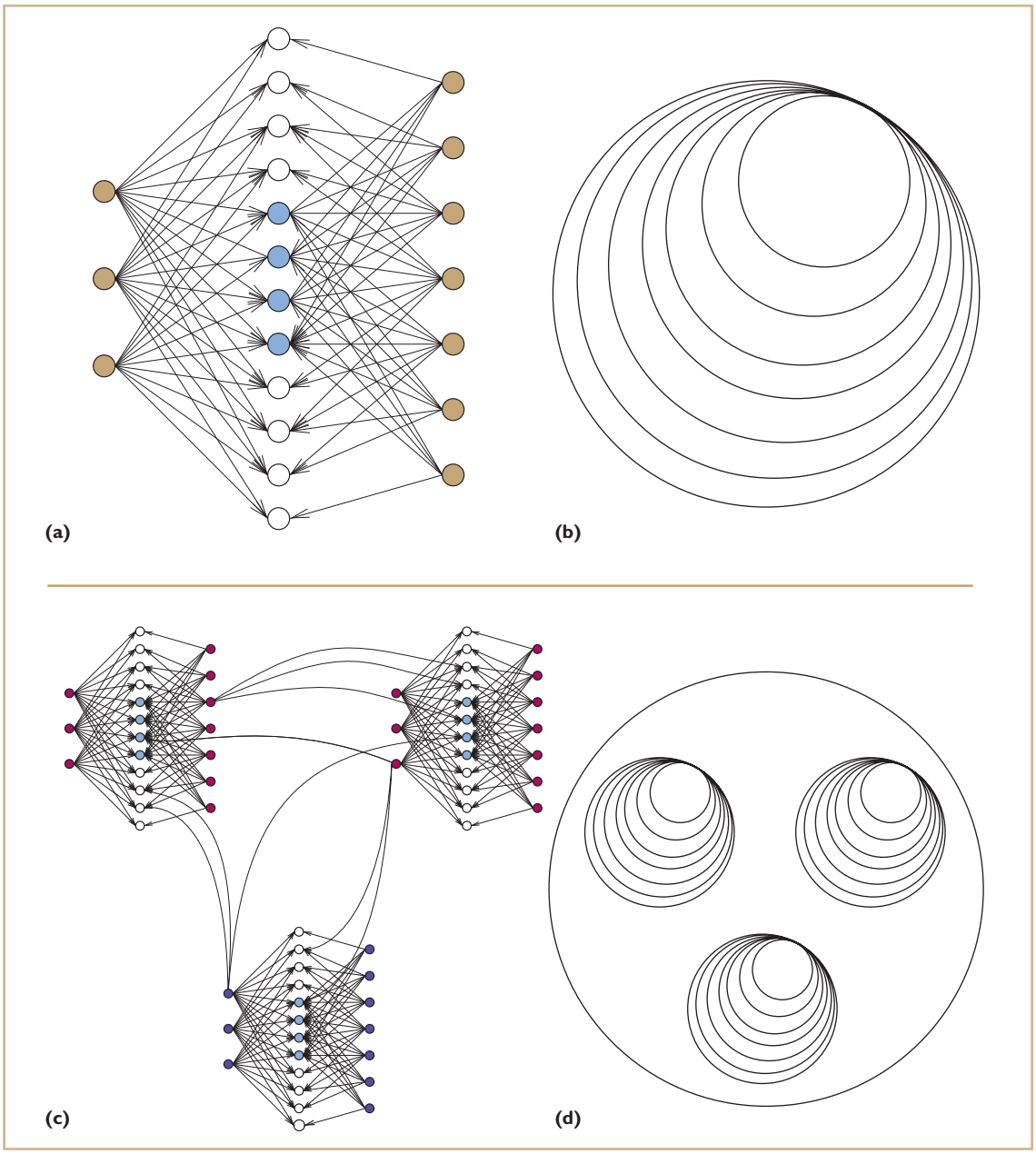

Figure 6. Two types of recommendation data sets involving straddlers. (a) A data set with a power law induces a low-risk scenario, (b) where increasing hammock widths cause a "nested clam shells" picture. Each circle in the Venn diagram denotes a group of people brought together. By increasing hammock widths, the sets become progressively smaller. (c) A dataset with power laws in only subgraphs and a few straddlers induces a high-risk scenario (d) characterized by the breakdown of a connected network into disconnected networks.

improvements in path length from 2 to 1 hammocks than, say, from 3 to 2.

Recommender systems typically require users to rate a minimum number of artifacts before they can make queries. This lets us look at the incremental benefit of providing additional ratings. As mentioned, the MovieLens data set comprised 943 people and 1682 movies, and we introduced a 944th person for this experiment, incrementally adding ratings to movies. After adding each rating, we computed the path lengths to four selected movies (*Star Wars*, *Tomorrow Never Dies*, *Robin Hood: Men in Tights*, and *Scream of Stone*) for various values of w. These movies were chosen to constitute a range from one that is usually rated most often (*Star of Wars*) to one that is rated least often (*Scream of Stone*). We then calculated the benefit using these values of l and w.

Figure 7 shows the benefit that a user can hope to get from the recommendations for the four movies. Each colored cell indicates that the particular benefit is possible for the corresponding number of ratings. The plot shows that a good recommendation for a less popular movie requires more ratings than for popular movies. A cell marked "infeasible" means that it is impossible to achieve the specified level of benefit with the given number of ratings.

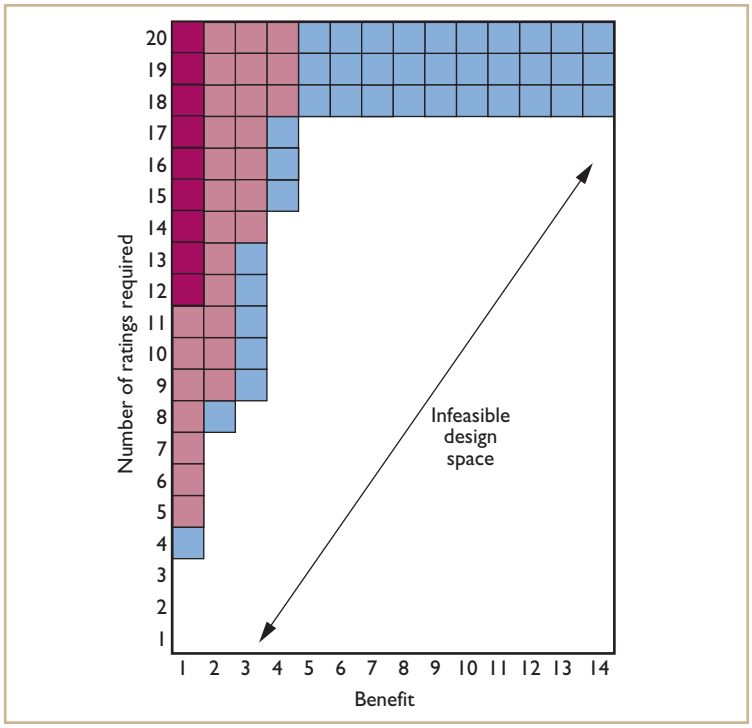

Figure 7. Ratings-to-benefit relationship. Less popular movies (red cells) require more additional ratings to improve recommendations than more popular movies (blue cells).

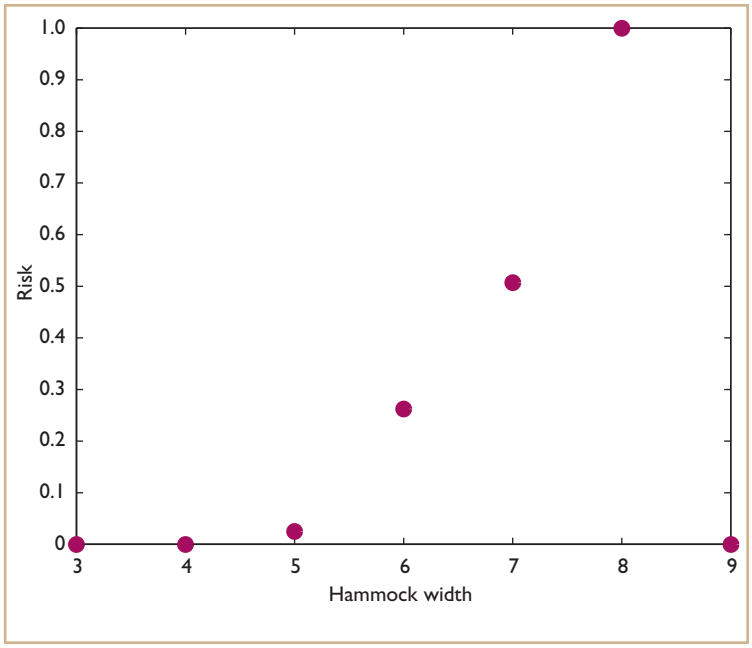

Figure 8. Risk as a function of hammock width. In our study, privacy risk rose rapidly until w = 8, at which point a single straddler remained. Risk dropped sharply to zero when the subgraphs disconnected at w = 9.

Modeling Risk

The privacy risk associated with recommendation systems is negligible in a power law dataset such as MovieLens, but as Figure 6 suggests, there is a risk to straddlers that connect subgraphs. The key intuition is that straddlers' ratings drastically reduce the distance between neighbors of the straddler node, and between their neighbors, and so on. Increasing the hammock width until the graph becomes three disconnected components, as in Figure 6d, reduces the privacy risk to zero. A hammock width of just less than this exposes straddlers and creates the highest degree of risk. We can thus model risk as the rate at which l decreases as a function of a parameter such as w:

$$\text{risk} = -\left(\frac{\partial l}{\partial w}\right)$$

To explore this setting, we created an artificial data set similar to the one in Figure 6c. The three subgraphs followed power law degree distributions and included 200 people and 75 artifact vertices each. Person nodes were each linked to 15 or fewer artifact nodes within the same subgraph. Specifically, the people and artifacts were ordered, and the b*th* person rated the first $\lceil 75b^{-e} \rceil$ artifacts. The value of e was calibrated to achieve a maximum rating of 15 artifacts for each person. We then added three extra people who rated 15 artifacts each across the three subgraphs (again with a power law distribution), and these became our straddlers.

The graph of connections became three disconnected components when $w = 9$. As Figure 8 illustrates, risk was highest when $w = 8$. Note that, unlike benefit, risk does not increase or decrease monotonically with w. Instead, risk rises rapidly until the point at which a single straddler remains, and then it drops sharply when the subgraphs disconnect.

It is not possible to provide a traditional benefit-risk profile for recommender systems because they aggregate ratings from many participants. The user benefits from "plugging into" the network of participants by providing a sufficient number of ratings. Risk, on the other hand, depends not only on what the user rates but on what other people rate. By logging in to a system, providing ratings, and

logging out, the user faces inherent risk — even without making queries. To gain benefit from the system, however the user must make queries.

Future Work

While a detailed solution for thwarting hackers is beyond the scope of this article, we can state some general guidelines. Like most problems in computer security, the ideal deterrents are better awareness of the issues and more openness in how systems operate in the marketplace. In particular, individual sites should clearly state the policies and methodologies they employ with recommender systems, including the role played by straddlers in their data sets and system designs. This is especially true for sites with multiple homogeneous networks (as in Figures 6c and 6d).

By conveying benefits and risks to users intuitively, recommender systems could achieve greater acceptance. We envisage three general ways to highlight the implications of our analyses.

- Present plots of benefit and risk versus user-modifiable parameters such as ratings, w, and l (if the algorithm allows their direct specification) to allow users to make informed choices about their levels of involvement.
- Qualify the risks and benefits associated with rating each individual artifact (as a function of the previous ratings in the system) as well as the extent to which a user becomes a straddler by each rating.
- Make recommendations involving straddlers only after enough others are involved to bring the risk to an acceptable level.

To gain the benefits of length reduction, sites could offer incentives for straddlers to participate. A plot such as in Figure 8 can help site designers choose an appropriate level of straddler involvement.

Current privacy management interfaces are woefully inadequate and their importance is only now being recognized.[8] Singh et al. have made a provocative comparison between community-based networks and recommender systems — namely, that people really want to control who sees their ratings and to know how recommendations are made.[9] Future research could extend our results to a distributed setting where users could be allowed to specify how data collected from their interactions are modeled and used.

References

1. J.L. Herlocker et al., "An Algorithmic Framework for Performing Collaborative Filtering," *Proc. 22nd Annual Int'l ACM SIGIR Conf.*, ACM Press, 1999, pp. 230-237.
2. C.C. Aggarwal et al., "Horting Hatches an Egg: A New Graph–Theoretic Approach to Collaborative Filtering," *Proc. Fifth Int'l Conf. Knowledge Discovery and Data Mining* (ACM SIGKDD 99), ACM Press, 1999, pp. 201-212.
3. D.E. Denning, P.J. Denning, and M.D. Schwartz, "The Tracker: A Threat to Statistical Database Security," *ACM Trans. Database Systems*, vol. 4, no. 1, March 1979, pp. 76-96.
4. B.J. Mirza, *Jumping Connections: A Graph-Theoretic Model for Recommender Systems*, master's thesis, Computer Science Dept., Virginia Tech, Blacksburg, 2001.
5. J.A. Konstan et al., "GroupLens: Applying Collaborative Filtering to Usenet News," *Comm. ACM*, vol. 40, no.3, Mar. 1997, pp. 77-87.
6. D.J. Watts and S. Strogatz, "Collective Dynamics of 'Small-World' Networks," *Nature*, vol. 393, no. 6, June 1998, pp. 440-442.
7. M.S. Granovetter, "The Strength of Weak Ties: A Network Theory Revisited," *Sociological Theory*, vol. 1, 1983, pp. 203-233.
8. T. Lau, O. Etzioni, and D.S. Weld, "Privacy Interfaces for Information Management," *Comm. ACM*, vol. 42, no. 10, Oct. 1999, pp. 88-94.
9. M.P. Singh, B. Yu, and M. Venkataraman, "Community-Based Service Location," *Comm. ACM*, vol. 44, no. 4, April 2001, pp. 49-54.

Naren Ramakrishnan is an assistant professor of computer science at Virginia Tech. His research interests include recommender systems, problem solving environments, data mining, and personalization.

Benjamin J. Keller is a visiting assistant professor of computer science at Virginia Tech. His research interests include recommender systems, symbolic computation, and software engineering.

Batul J. Mirza is a research associate in the computer science department at Virginia Tech. Her research interests include recommender systems, graph theory, and social networks. She received an MS in computer science from Virginia Tech.

Ananth Y. Grama is an associate professor of computer sciences at Purdue University. His teaching and research interests include parallel and distributed algorithms, large-scale data handling and analysis, and scientific computations.

George Karypis is an assistant professor in the computer science and engineering department at the University of Minnesota. His research interests include data mining, bioinformatics, parallel algorithm design, and information retrieval.

Readers can contact the authors at naren@cs.vt.edu.