-4mm

Eurographics/ IEEE-VGTC Symposium on Visualization 2010
G. Melançon, T. Munzner, and D. Weiskopf
(Guest Editors)

Volume 29 (2010), Number 3

SmallWorlds: Visualizing Social Recommendations

Brynjar Gretarsson†, John O'Donovan‡, Svetlin Bostandjiev§, Christopher Hall¶, Tobias Höllerer||

[2]Department of Computer Science, University of California, Santa Barbara

Abstract

We present SmallWorlds, a visual interactive graph-based interface that allows users to specify, refine and build item-preference profiles in a variety of domains. The interface facilitates expressions of taste through simple graph interactions and these preferences are used to compute personalized, fully transparent item recommendations for a target user. Predictions are based on a collaborative analysis of preference data from a user's direct peer group on a social network. We find that in addition to receiving transparent and accurate item recommendations, users also learn a wealth of information about the preferences of their peers through interaction with our visualization. Such information is not easily discoverable in traditional text based interfaces. A detailed analysis of our design choices for visual layout, interaction and prediction techniques is presented. Our evaluations discuss results from a user study in which SmallWorlds was deployed as an interactive recommender system on Facebook.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Interfaces and Presentation]: Information Storage and Retrieval—Information Search and Retrieval

1. Introduction

In this paper we introduce SmallWorlds†, a visual interactive tool which generates accurate and useful item recommendations based on a combination of Facebook user profile data and user interactions with a simple graph-based interface (See Figure 1). What sets SmallWorlds apart from other recommendation systems is that the novel visualization/interaction strategy is designed to work within the constraints of the Facebook API. That is, the system does not require large numbers of users in order to produce good recommendations– Facebook contains over 400 million profiles but the Facebook API does not support unauthorized reading of item preference information beyond the immediate friend group [Fac], making traditional item recommendation strategies such as automated collaborative filtering [RIS*94] [HKR00] largely redundant on this database.

Standard profile-similarity metrics used by traditional recommendation systems are not sufficient to produce good recommendations in small datasets. The shortcomings of techniques such as ACF have been well documented in CF literature [RIS*94]. Given the data privacy restrictions imposed by the Facebook API, by default we have only a small set of item preference data from which we can compute recommendations. However, we believe that since this data has been pre-filtered by peers with a direct social connection to the active user, it can be used in a visual application to produce satisfactory recommendation in spite of sparse profile overlap.

In this paper we evaluate our interactive visualization with a user study which contributes results at the intersection of visualization research and recommender systems and an additional result for the computation of recommendations. These results stem from the following research questions: Firstly, are visualizations important for recommendation systems and do they provide transparency and increase user trust in the system's prediction? Secondly, is interaction important in the visual interface, and does it give the user a sense of control over the underlying algorithms and final output. Thirdly, by visualizing the computational process of generating recommendations, does a user's pick up "ambient information" about his peer's tastes' in general [War04]. Lastly, can pre-existing social connections [Mil67] be used in addition to profile similarity to boost performance of a recommender system? What are the effects of this both in terms of user satisfaction and accuracy?

2. Related Work

Background research for this work is presented in three parts. Firstly, a discussion of the state of the art in visu-

† brynjar@cs.ucsb.edu
‡ jod@cs.ucsb.edu
§ alex@cs.ucsb.edu
¶ chall01@cs.ucsb.edu
|| holl@cs.ucsb.edu
† http://apps.facebook.com/smallworlds

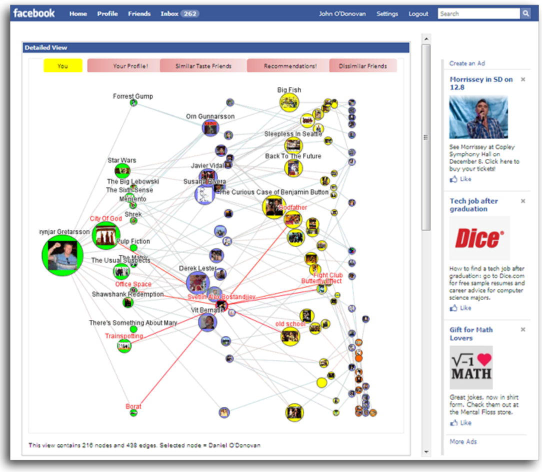

Figure 1: *Screenshot of the SmallWorlds Facebook Application.*

alization design and research for graph and network data is presented, focusing on web-based approaches. Following this, a brief overview of collaborative recommender systems research is presented, as this is required for understanding of the relevance of the novel visualizations presented later. An analysis of a range of visual interfaces for systems that produce recommendations is presented, with a particular focus on those which are deployed as rich internet applications. Recently, much attention has been given to visual interfaces for recommendation systems by the AI community, e.g. [ZJP08, MKK06, HKR00] but this interest has not been matched by the visualization community. Our background research focuses on the work at the intersection of these two research communities.

2.1. Web-based Graph Visualization

Designing a representation for a complex network that is easily comprehensible yet still comprehensive is a non-trivial task. For large scale data visualization, Ben Shneiderman's visualization mantra of "overview, zoom and detail on demand" has been popular over the years [Shn96]. However, recent graph visualization systems such as Vizster [HB05] and TreePlus [LPP*06] concentrate on navigation and information discovery only in a subset of the broader data universe, and do not provide a big-picture overview. The design paradigm in these systems is that users should always have not only a comprehensible, but a fully *readable* data representation on the screen at any given time. In a similar manner to PivotGraph [Wat06], WiGis [GBOH09] and many other graph visualization systems, e.g. [SMO*03] [BM98] and [Tou], SmallWorlds leans towards Shneiderman's approach in that overview, zoom, panning and zoom-based detail are important design considerations. Small world graphs [Mil67] are common in today's social networks and are notoriously difficult to visualize as a whole. Our application focuses on a user-specific perspective of a small world graph, in a similar manner to the Natto view in [**?**]. More recent work by Trethewey [**?**] uses topology-based interpolation of user interactions to attain node-specific views of large graphs.

As with all applications on Facebook, functionality for SmallWorlds must be supported in a standard web browser. Traditionally, applications that visualize graph and network data have been desktop based. For example, Cytoscape [SMO*03] [Cyt], Pajek [BM98], and some implementations of Tom Sawyer Visualization [Sof09]. Over the past few years, some graph visualization systems have moved towards rich internet architectures (RIAs) capable of providing interactive and responsive graph interfaces through a web browser. Examples of such applications include Touchgraph [Tou] and IBM's Many Eyes [DVWK08], which use Java Applets to support web functionality, and Tom Sawyer Visualization [Sof09], which has multiple web-based implementations. Interactive recommendation graphs in SmallWorlds are presented natively in a web browser, since this approach has been shown to be more scalable and less dependant on limiting technologies than other web-based approaches [GBOH09].

2.2. Recommender Systems

Recommender systems have been developed as a solution to the information overload problem and have been a focus of research within the AI community for close to twenty years [SM95]. The overarching goal of a recommender is to predict the right item to the right person at the right time. They can be implemented as basic content-matching systems [MMN02] [GRGP01], which simply match textual descriptions of recommendable items to a description in a user's preference profile. Such content-based recommendation techniques are generally considered as a rudimentary class of recommender and historically they suffer from problems such as narrowness, since they can only recommend items similar to what is already in a users profile, they also fail on non-machine analyzable data such as humor for example. The most widely used technique for generating recommendations is Automated Collaborative Filtering (ACF) [BMZB05][Kar01][OWS02][RIS*94][SKKR01], which is the technique we focus on for the SmallWorlds visualizations. ACF attempts to model the normal social process of asking a friend for a recommendation. In brief, ACF algorithms compute a neighborhood of users based on some correlation function (usually Cosine or Pearson's Correlation [OWS02] over vectors of rating data) and use that neighborhood to predict items that a user has not yet seen and that have been "liked" by his/her closest neighbors.

Through visualization we are creating an "explanation interface" for our recommender system. Some research has been carried out into the effects recommendation explanation has on the overall user experience with the system. A prominent work in this field is Herlocker's study of recommendation explanations [HKR00]. Herlocker et al. evaluate a "white box" conceptual model of recommendation as opposed to the run-of-the-mill black box approach. They present a user study where 21 different recommendation interfaces are presented to users, explaining various types of internal information from the recommender algorithm. Their general findings agree with Middleton's [Mid02], in that in that "explanation interfaces lead to improved acceptance of a predicted rating." Furthermore, previous work by the authors in [OSG*08] focused on the visualization of genre information to elicit preference-feedback from users to enhance the quality of movie recommendations generated from a large scale data set.

In this work, we are interested in a visual interactive interface for an ACF algorithm which is designed to work on the popular social networking site Facebook. Up until recently, it was possible to extract large scale item preference data from the Facebook web interface since profile access was granted by default to members of a users broad Facebook network. Work in [WBS*09] has attempted to capture this information for the purpose of large-scale analysis. However, new privacy policies of December 2009 [Fac] have restricted profile access only to those directly specified by the profile owner. Since the Facebook web interface and API only permit an application access to item preferences of those profiles in the immediate friend network, we are very limited in the amount of profile information we can use to generate recommendations. This constraint poses an interesting research problem however: Given that we can only see a friend group, can we design an algorithm and visual interface that will produce quality recommendations that will satisfy users? In our evaluations we demonstrate that by providing a user with a visual explanation and interactive control mechanism for an ACF algorithm, and by combining traditional overlap in preference profiles with pre-existing social connections, we can achieve this goal to a reasonable degree.

3. Visualizing Taste Spaces

The Facebook API provides access to a user's friends and to their item preferences. Given this constrained information space, we have designed an interactive visualization that produces real-time item recommendations in an open and transparent manner. We now discuss a high level overview of the layouts and functionality of the SmallWorlds application. This is followed by a more detailed description of the layout and interaction algorithms used. Figure 2 presents an overview of a sample graph from the SmallWorlds application. The visualization is essentially a second-order graph which links user nodes through the items which they have in common. Items generally take the form of books, movies and music, but can be extended to include any field from the Facebook API.

The key design in Figure 2 is a constraint which separates users and items into distinct layers. Layers are defined by the $b_{i,j}$ values in Figure 2.

- *Layer 1* - The active user's node.
- *Layer 2* - The active user's profile items.
- *Layer 3* - Friends who have items in common with the active user.
- *Layer 4* - Items that are not in the active user's profile but are liked by friends in layer 3, i.e. the candidate recommendation set.
- *Layer 5 and subsequent layers* - Friends who have no items in common with the active user and items in their profiles, but not in the profiles of friends in layer 3.

It is important to note that nodes are constrained within their initial layer and do not move across the $b_{i,j}$ lines. The first layer contains a node representing the active user who is logged in to the system. This node is locked in position and remains static through all layouts and interactions. The second layer shows the items in the active user's profile. Because we do not have explicit scaled ratings on Facebook, and only a binary item presence, in the initial layout, layer two items are all positioned equally in a vertical vector, indicating equal preference for each by their presence on the Facebook profile. A user can interact with these nodes afterwards to express granular preferences for individual items by dragging them towards or away from their avatar node. Figure 2 shows layer two items after they have been interacted with. The items shown in this example are music artists, however, the graph can contain multiple item types, such as books and movies for instance, which obviously drastically alters the final recommendation set.

Layer three contains those friends who have items in common with the active user. For the initial layout, the proximity of a layer three node to the boundary $b_{2,3}$ is governed by the amount of taste overlap that friend has with the active user. Such nodes are automatically scaled up to illustrate their importance in the recommendation process. For example, in Figure 2, profile x_{3a} has more items in common with x_1 than any other node in the graph. Hence the node is drawn close to the boundary $b_{2,3}$.

Layer four can be considered the "important" layer, as it contains the candidate recommendation set. These are items which are not in the active users profile but are in the profile of users in layer three. Again, these nodes are placed horizontally between the layer boundaries ($b_{2,3}$ and $b_{3,4}$) based on the number of edges incumbent from layer three nodes. Nodes closer to the $b_{3,4}$ line are more popular and are scaled, highlighted and labeled, because they represent the output of our visual recommender system. If the active user finds an item in layer four that she likes, then she can drag it towards the active user's node and it will be added to her profile and put with her other items in layer two.

Layer five contains friends who do not have anything in common with the active user, but have some items from layer four in their profile. Layer six would then contain these friends' items which are not liked by any friends in layer three and so on until all friends and items are represented. Due to the lesser importance of layer five and beyond for the recommendation process they are given much less space in the visualization than the first four layers, as can be seen in Figure 1 and Figure 2. Figure 3 shows an alternate visualization of the recommendation graph, where the layers are bounded by concentric circles instead of areas bounded by straight lines as in Figure 2.

3.1. Interaction and Weighting

Now that we have discussed the functionality of each layer in our visualization, we move on to our weighting and interaction model. In this section we present many formulas that are used in the process of computing the recommendations and determining the position and size of the nodes in the graph. We must first define our variables which are used consistently across all the formulas in this section. I is the

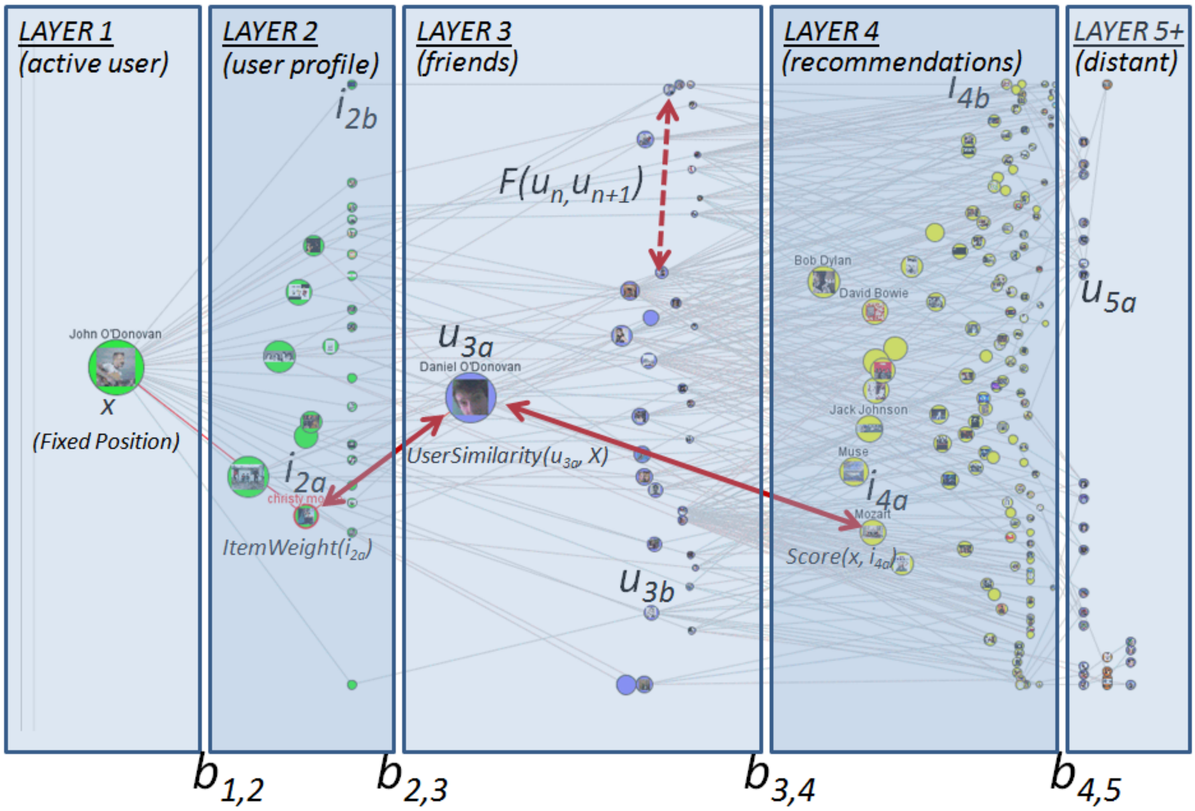

Figure 2: *The SmallWorlds recommendation interface for an active user x. Node i_{2a} represents an item in the active user's profile. Position of this node within layer 2 determines a preference weighting $ItemWeight(i_{2a})$. Node u_{3a} represents a friend who has items in common with the active user. Position of this node within layer 3 is computed by $UserSimilarity(u_{3a}, x)$. The dashed line $F(u_n, u_{n+1})$ represents a repelling force between nodes, present in all layers. Node i_{4a} represents a recommended item with a position within layer 4 computed by $Score(x, i_{4a})$. The remainder of the active user's friend nodes are placed in layers 5 and higher based on their graph distance from the active user's node. The solid red arrows give an example of the effects of moving item nodes in layer 2, such as i_{2a}.*

set of items in the graph, U is the set of users in the graph, excluding the active user, which is referred to as x. i is used to refer to an item from I and u refers to items from U.

Each item i in layer two (the active user's profile items) holds a weight between 1 and 5 (both inclusive). The default weight is 1 for every item to simply reflect presence of an item in a profile. Any item node can be clicked on and dragged horizontally, towards or away from the active user's avatar. Based on the drag distance, an associated weight changes until it reaches a maximum of 5. This weight does not need to be an integer value; it can be any value in the range between 1 and 5. This simple item weighting is given by Equation 1:

$$1 \leq ItemWeight(i) \leq 5 \tag{1}$$

However, if the user drags the node outside the boundary between layer 2 and layer 3 (beyond the position for ItemWeight = 1) then the selected item is removed from the user's profile and placed in layer 4. The user can also drag that item, or any other items in layer 4, back into layer 2 to add it to his/her profile. This gives the user the ability to easily edit their profile within SmallWorlds. Of course, the weight given in Equation1 is just the weight for a single item. In order to compute the positioning for users in layer three, we need to know the total weight of all of the items that a layer three user has in common with the active user. Nodes with larger weights are placed closer to the left border and increased in size to illustrate their relevance, as with profile x_{3a} Figure 2. This computation forthe active user x and a given user u follows as Equation 2:

$$\begin{aligned} &TotalWeightOfCommonItems(x,u) = \\ &\mathop{}_{i \in I} (Likes(x,i) \cdot Likes(u,i) \cdot ItemWeight(i)) \end{aligned} \tag{2}$$

where $Likes(u,i)$ returns 1 if item i is listed in the profile of friend u and 0 otherwise.

In our initial design, this value was computed as a count of the number of common items between the two users. However, Equation 2 reflects user preference more accurately as it considers the preference tweaking that a user makes on their profile items. Since each graph on Facebook is different and we do not have a predefined dataset and since users can modify weights through interaction with the visualization, we must calculate the total item weight on the fly in order to gain meaning from the value in Equation 2. The total weight of items for a user u is given by Equation 3.

$$TotalWeightOfItems(u) = \mathop{}_{i \in I} (Likes(u,i) \cdot ItemWeight(i)) \tag{3}$$

Friend nodes in layer three of Figure 2 are also initialized with a default weight of 1. If a friend node is moved further from the active user's node on the left, the weight becomes less than 1, and conversely is increased as it is dragged closer. This value is represented as $UserWeight(u)$ (similar to Equation 1). The range of this function is bounded by 0 on the lower end, but only bounded by the border between layer two and layer three on the upper end, i.e. the user can keep dragging the node towards the active user's node (and thus increasing the $UserWeight$) until it reaches the border between layers two and three. Dragging a friend who has little in common with the active user all the way towards the layer two boundary will result in a much higher $UserWeight$ than dragging a friend who has much overlap to the same position, although their influence on the recommendation will be the same.

Putting this all together, we get a value for the similarity of a given friend, based not only on overlap between profile items, but also based on user interaction, which we consider to be highly important as it can be used to express facets of trust, current mood and other user-specific preferences. The similarity of a given friend node u to the active user x is represented as Equation 4.

$$UserSimilarity(x,u) = \frac{UserWeight(u) \cdot TotalWeightOfCommonItems(x,u)}{\sqrt{TotalWeightOfItems(x) \cdot TotalWeightOfItems(u)}} \tag{4}$$

However, if a layer three friend node is moved up to a boundary line, thus expressing the maximum preference for that friend, the above computation can yield an inconsistency if the user then expresses a higher preference for one of that friends's items. This produces an "over the top" effect where the user's preference is higher than the maximum for our scale. To correct this, we place a limit on the upper bound for the total similarity. This is shown in Equation 5

$$BoundedUserSimilarity(x,u) = \min(1, UserSimilarity(x,u)) \tag{5}$$

To recap, layer four contains items which exist in the profiles of friends in layer 3 but are not contained in the active user's profile. Item nodes in this layer are scaled and positioned based on the number of edges that link to layer three (similar friend) nodes, and are scaled based on the respective weightings of those connected layer three nodes. Thus, the items in layer four are the candidate items for recommendation, and the larger nodes which are close to the boundary $b_{3,4}$ constitute a list of top-n recommendations. The SmallWorlds graph visualization can be viewed as a visual, graph-based adaptation of a user-based ACF algorithm, as proposed by Resnick in [RIS*94], with one distinction–Facebook preference data is binary. Accordingly, the "candidate" items in layer four are ordered based on Equation 6, where Likes(u,i) returns 0 or 1 (user u likes item i, replacing the explicit numerical rating used in [RIS*94])

$$Score(x,i) = \mathop{}_{u \in U} (Likes(u,i) \cdot BoundedUserSimilarity(x,u)) \tag{6}$$

The interaction model discussed here produces dynamic item recommendations in real time based on the data accessible through an individual Facebook profile. The system is accessible through a simple, web-based graph interface which is described in the next section.

Apart from the weighting and interaction models, several additional forces are applied to the nodes to produce the clean layout shown in Figure 2. At a general level, the graph layout can be viewed as a layer-constrained Fruchterman-Reingold force directed algorithm. The basic algorithm has been substantially altered and a number of constraints have been added. Each node has a preferred distance from the central node (based on $ItemWeight$ (layer 2), $userSimilarity$ (layer 3), and $itemScore$ (layer 4)). Every 5th iteration of the F-R based layout algorithm, the node is pulled back to this distance from the active user's node. In the case of circular layout3 the actual euclidean distance is used, but in the example in Figure 2 a 1-dimensional distance along the x-axis is used. Lastly, nodes are spread out along the vertical axis by a repelling force between each node (a constrained version of the standard F-R layout algorithm). In the evaluated version of the system, node size was not a factor in the layout algorithm, although this has been incorporated into the current implementation of the system. While F-R is not the most efficient layout algorithm, it was deemed sufficiently fast for our recommendation graphs with relatively simple constraints ($u_{xpos} = x_{xpos} + a$). For better scalability and more complex constraints, Dwyer's layout method, presented in [Dwy09] would be a better choice.

3.2. Web-Based Architecture

Before we present our evaluation, we must address several important points about the architecture of the SmallWorlds application, which delivers real-time, smooth graph interactions and animations in the web. To achieve scalability, fast graph layouts, and real time interactions, all of which potentially require a lot of computational resources, it is not always possible to rely on the capabilities of a client machine. Accordingly, a design choice was made to avoid using standard rich internet technologies such as Java Applets and Flash, as these technologies rely on client-side processing.

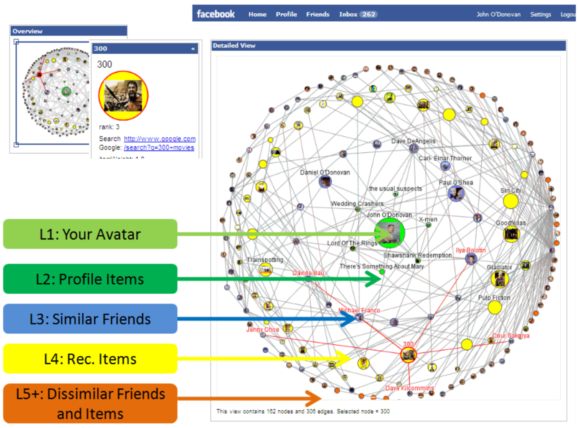

Figure 3: *Example of a concentric layout for item recommendation. As profile or friend nodes are dragged, item nodes move relatively based on the weighting and interaction models.*

The SmallWorlds Facebook application is deployed as an I-Frame in the standard Facebook application architecture‡. Within this frame, the SmallWorlds graph is simply a bitmap representation of a graph model that has been computed on a remote server and streamed across a network in real time (approximately 10 to 20 FPS for a standard network). Interaction is achieved on this graph by capturing mouse movements in JavaScript and streaming them back to a graph server, where the appropriate layout or interaction functions compute a new representation of the graph model and stream it back to the server. Research on this web-based visualization technique [GBOH09] has shown that given a reasonably fast interaction algorithm, responsive interactions are supported for graphs of the order of hundreds of thousands of nodes§. An advantage of this architecture is that no browser plug-ins are required, and performance of layout and interaction does not depend on the potentially poor capabilities of a client's computer. Additionally, this framework supports all the parts of the information visualization mantra[Shn96] by initially showing a view of the entire graph, then enabling the user to zoom into any parts of interest, and finally allowing the user to request more details about selected nodes. For the Facebook profiles that have been analyzed thus far, since we have only tested with books, movies, and music in a single graph, the largest size graph we have visualized for a single profile is still less than 2000 nodes. However, if the application amasses enough users, the architecture is capable of supporting fast interactive visualizations with a multitude of domain items on a single graph.

4. Evaluation

Since the SmallWorlds application is a visual interactive tool, deployed over the limited dataset from Facebook API, we cannot adopt the standard large scale automated tests to evaluate the quality of recommendation algorithm in terms of accuracy. The nature of the application also precludes standard Precision-Recall testing used in Cranfield-like evaluations. Accordingly, we test the quality of the interface through analysis of interactions in a user study. Although they are highly dependent on each other, a distinction should be drawn between the qualitative evaluation of the *visual interface* and the *recommendations* it is used to produce. For example, we find that users with more than 200 friends reported a much higher satisfaction rating with the system, compared with users who had less than 50 Facebook friends. This most likely occurs because they had much richer graphs due to increased taste-overlap with their many friends. To test the quality of recommendations produced, we compare SmallWorlds against a benchmark system which uses a database of millions of ratings [MAL*03], and perform a small scale leave-one-out accuracy comparison.

4.1. User Study

To evaluate the interactive visualization components of the system, we performed a study consisting of 17 controlled user trials. The objective of the trials was to address the research questions raised in the introductory text. The user study was designed to gain as much information as possible about users' overall experience while using the system to get recommendations and explore their peer's movie tastes. Specifically, the study looked at sense of control, and satisfaction levels. In addition, the study examined the degree of "latent information gain" that occurred while users interacted with the visual interface. A comparison was made over a range of tasks between the standard, html-based Facebook interface, and two configurations of our SmallWorlds interface. Lastly, the study compared results from the SmallWorlds system to a benchmark recommendation system in terms of user satisfaction. The purpose of the comparison against the text-based interface is not to show that SmallWorlds is "better" than the Facebook interface, but rather that it does have inherent shortcomings because it is largely text-based. For example, discovery of information about trends, cliques and other metrics that require analysis of data from multiple friends profiles simultaneously.

The study opened with a pre-test questionnaire, followed by a set of 7 information discovery tasks involving a benchmark recommendation system, the standard Facebook web interface and various configurations of the SmallWorlds application. On average, studies lasted for 30 minutes, the majority of which was spent on the 7 tasks. To eliminate bias, question ordering was randomized and questions which required Likert Scale (1-5) answers had the scale direction randomly switched. Orders of tasks were also randomly switched to minimize possible learning effects which could potentially skew results. User interactions were logged and initial graph visualizations were stored for later analysis. After completion of each task, users filled out a post-study questionnaire designed to compare the interfaces, and another questionnaire was filled out shortly afterwards to further analyze the SmallWorlds tree-layout which was the best overall performer in the earlier test.

‡ developers.facebook.com
§ www.wigis.net

4.1.1. Participants

17 participants undertook the study. Participants were mainly from a university at all levels from undergraduate to faculty. The group of participants consisted of 14 males and 3 females, ranging in age from 21 to 33 with an average of 27.2 years (median: 27). Most participants reported that they were regular Facebook users (23% daily, 60% weekly). Participants exhibited diversity in number of Facebook friends, which ranged from 50 to 1200 with an average of 240 (median 215). The most commonly reported favorite Facebook application was online poker. Quiz applications were also popular. We queried about data privacy on Facebook, participants almost unanimously reported that they would not permit an unfamiliar application to access their profile data. On average, participants reported that they were familiar with recommendation systems and have used them in the recent past. When asked about where participants get recommendations for new media such as music and movies, the large majority (73%) reported that it was from online sources as opposed to direct communication with friends. To prevent studies from becoming over-lengthly, our data domain was restricted to movies only. It was ensured that all participants had a minimum of 10 movies in their Facebook profile prior to the study. Profile size ranged from 10 to 25 with an average size of 14 (median 15). Reported knowledged of the domain varied from poor to excellent across all participants with the majority reporting a very good knowledge.

4.1.2. Information Discovery Tasks

A set of 6 tasks (tasks 2-7, below) were designed to allow for a comparison between the Facebook interface, a SmallWorlds tree-like layout and a SmallWorlds circular layout. The list below presents an overview of each task. A 2-task-3-interface Graeco-Latin square approach was used to assign tasks to users to avoid a learning pattern. All users completed task 1.

1. *Task 1:* Familiarization (5 mins, supervised)
2. *Task 2:* Find popular items in your peer-group.
3. *Task 3:* Find your 3 most similar peers
4. *Task 4:* Find your 3 least similar peers
5. *Task 5:* Get recommendations through layout only
6. *Task 6:* Get recommendations through layout and interaction
7. *Task 7:* Get recommendations through layout and interaction, with layer 4 (candidate-set) items hidden.

Figure 4 shows the results from the comparison study. The S_n values on the x-axis represent the statements shown in Table 1, which users were asked to rate on a 5-point Likert scale of increasing satisfaction. A within-subjects ANOVA revealed a significant difference ($p < 0.05$). As expected there was a significant improvement reported over all metrics ($S6$ being negative) for the two visual interfaces over the standard Facebook interface. To re-iterate, the Facebook text based interface is included in the comparison only to highlight that it does have shortcomings for certain complex information discovery tasks such as those listed above, and that it could benefit from the *addition* of a visual component. The tree-based layout (SW-Tree) produced better satisfaction ratings than the circular layout (SW-Circular) with the exception of $S5$ in which the circular layout was reported to

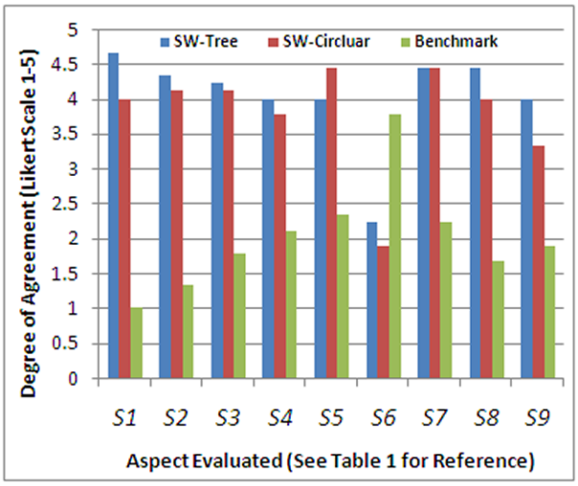

Figure 4: *Comparative results showing user perception of three interfaces.*

be around 12% more intuitive. This was an unexpected result since it is easier to visually distinguish between node layers in the tree layout. Across the other metrics listed in Table 1 the increase in satisfaction for the SW-Tree method over the circular method ranged from about 6% to about 12%. When asked if the interface was "clumsy" for the given tasks, there was disagreement for both tree and circular layouts ($S6$) and agreement for the text-based interface. Participants strongly agreed that the tree-graph was globally effective for finding commonalities in movie taste ($S1$), was highly informative ($S7$, equally with the circular graph) and generally helped to explore the given topic ($S8$). The largest win for SW-Tree over SW-Circular was on $S9$, which refers to profile building capability. On the facebook interface, during profile building, items are suggested in drop-down lists as they are typed into a profile, but it is not possible to look at a friend's profile and click an item to have it appended to your profile. This can only be performed with a standard copy-and-paste, which involves further navigation and reloading of web pages. Profile-building is facilitated in SmallWorlds based on items in friends' profiles. A user can click on any item that is liked and drag that item all the way from layer four (friends items) into layer two (the active user's profile). This action causes the dragged item to be added to the active user's Facebook profile, and causes the weighting model to rearrange the graph layout based on the updated profile information. It is believed that this feature caused the preference to lean towards SW-Tree for $S9$, since the layer boundaries (b_n values in Figure 2) are less clear in the SW-Circular method (shown in Figure 3).

The results shown in Figure 4 clearly show a significant preference for SW-Tree over the other approaches. Once this result was ascertained, users were given a more detailed questionnaire relating specifically to the SW-Tree interface. This second questionnaire was designed to gain more inforamtion about the user-experience with the SW-Tree interface, and to probe into user satisfaction with the recommendations generated by interactions with the SW-Tree vi-

#	*QuestionDescription*
S1	effective for finding commonalities in taste
S2	item popularity easily discoverable
S3	interesting items easily discoverable
S4	was easy to use
S5	was intuitive overall
S6	was clumsy overall
S7	was informative overall
S8	helped you to explore the given topic
S9	helped you to build your movie profile

Table 1: *Likert Scale Questions from the User Survey.*

sualization. Figure 5 shows both the questions asked and the reported results for each. Again, questions were posed in a 5-point Likert scale format. A one-way ANOVA showed significance with $p < 0.072$ on average. Both the first and last questions in Figure 5 are repeated from the previous questionnaire with a different wording as a sanity check. Ratings only differ from their partner questions (the leftmost bar of $S6$ and $S1$ in Figure 4) by less than half a rating point (with one of the results exactly equal). It was found that users generally liked the interface and felt that they would like to use the system again on their own time to explore their peers' item preferences. Participants were asked about the sense of control that the visualization/interaction gave them over the item prediction algorithm. The result reported (Q4 in Figure 5) is less than originally anticipated at 3.93. This is most likely because of the item nodes in layer 4. In our initial design, layer four nodes, which are the set of items in friend's profiles which are not in the active user's profile, were made non-interactive. That is, they can not be moved individually with a click-and-drag. The original motivation for this decision was to make the SW-tree layout model a traditional ACF algorithm as closely as possible. In ACF, predictions are generated based on user ratings on their profile items, not necessarily on the candidate recommendation set. However, freetext comments at the end of the survey included the following "a sense of restriction", "wish I could move all of the nodes", "last layer nodes should be movable". Accordingly, the layout algorithm and interaction model was updated shortly afterwards to extend weightings to layer four items, and allow users to rate them. This also provides a good feedback mechanism to assess system performance. If many individual layer four items are dragged away, then the prediction algorithm is doing badly for the current user.

Probably the most important result from this questionnaire shows us that users felt that the complex process of recommendation was explained to them as they were interacting with the SW-Tree layout. This sentiment is manifested in result 8 of Figure 5 as about 4.5 from a possible 5 rating points, and is a very promising result. If a complex process such as ACF can be easily explained through an interactive graph interface, then the SmallWorlds tool can serve as a pedagogical aid for ACF and possibly for other collaborative algorithms. Moreover, by explaining where recommended items are coming from, we are increasing the trust and confidence that end-users will place in the final output.

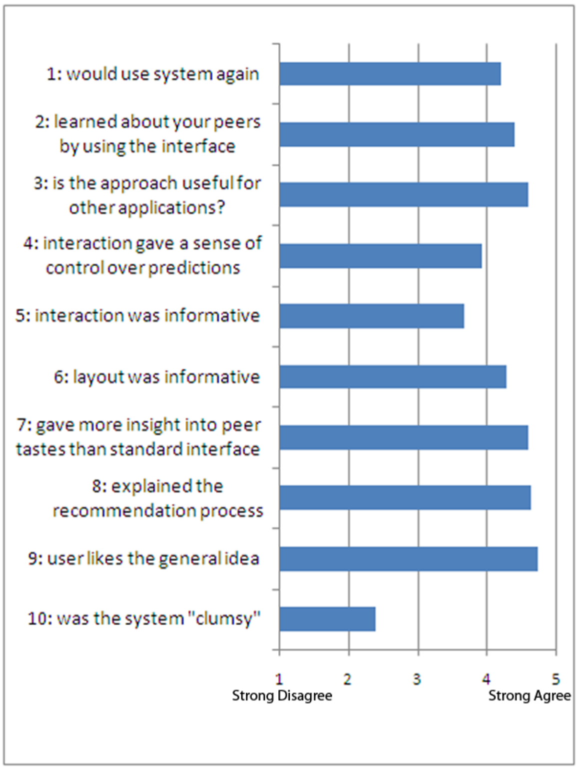

Figure 5: *Overall user experience with the Smallworlds tree-like graph.*

4.2. Quality of Predictions

Because of the novel nature of the SmallWorlds application, a fair comparison with a benchmark is difficult to compose, since benchmark recommendation systems such as MovieLens [RIS*94] [HKR00] are usually run on larger datasets. Although accuracy of a prediction system is not a common focus among visualization researchers, the authors feel that it is important to demonstrate the quality of predictions which are generated from interactions with the SmallWorlds visual recommender, as they have a bearing on the overall experience that users had with the visual interface.

4.2.1. User Satisfaction

Since our system only operates on the limited data available from Facebook, and since our system requires user tweaks and interactions, it is difficult to compare with MovieLens directly. However, during the user evaluations, the Facebook movie profile was stored (with permission) from every user and input into an implementation of a benchmark ACF prediction algorithm using the MovieLens 100000 numerical rating dataset [MAL*03]. An ordered list of 10 recommendations was generated by the algorithm for each participant. These were randomized and the participant was asked to rate them. Interestingly, when compared against the satisfaction rating generated from SmallWorlds which used only hundreds of binary ratings, made from close friends of the target user, the empirical accuracy difference was surprisingly

Method	MovieLens	SW-Tree (interactive)	SW-Tree
Satisfaction	4.25	4.19	3.78

Table 2: *Satisfaction ratings of item predictions for Movielens and for SmallWorlds with and without user interactions.*

small, especially after interaction occurred in the interface. Table 2 shows the results of this comparison for the benchmark and the SW-Tree method both with and without user interactions. By using the SW-Tree layout which is computationally equivalent to a standard ACF algorithm, but only uses peer profile data from Facebook, a mean absolute error (MAE) of 1.22/5 (or 24%) was produced. When interaction was applied (without the user seeing the candidate recommendation set), MAE was reduced to 0.81 (16.2%). Considering that the MovieLens system, which used several orders of magnitude more training data produced an MAE of 0.75 or 15%, only 1.2% better than our SW-Tree approach, we extrapolate that the use of underlying social connections can greatly boost the performance of an ACF algorithm. Furthermore we find that a visual interface can increase both accuracy as well as confidence in the system's predictions.

Since participants in our user trials had varying numbers of Facebook friends, a brief analysis examining the correlation between this neighborhood size and satisfaction with the system predictions was performed. While the analysis does incorporate quality of predicted items, it also considers the various different graphs that were created based on participants friends' profile items. In general, participants with more friends tended to have more overlap in taste, that is, more of the items in their profile overlapped with friends' items. Therefore, more layer two nodes appeared on their SW-Tree visualization. Figure 6 is a scatterplot with a trend line showing the number of friends on the x-axis and overall satisfaction with the system on the y-axis. The data series represent reported satisfaction using layout only, and then using layout and interaction. There is a clear improvement in user satisfaction shown for most cases after interaction has been performed. Furthermore, an interesting resuls is that the analysis shows a tentative ROC-curve which highlights 200 friends as a threshold number for a participant to report a high level of satisfaction with the recommendations. Participants with more friends than this threshold do not exhibit a relative increase in satisfaction.

4.2.2. Automated Accuracy Test

To boost the reliability of our previous results, which were based only on 17 participant profiles, we performed a crossfold validation through a leave-one-out analysis based on the profiles from the user study. For every profile gathered in our study we ran both SmallWorlds and MovieLens QuickPick[Gro09], with one item removed from the profile, until each item had been removed in turn. The top 12 recommendations predicted by both systems were recorded using the test profiles. For this automated test, no interactions were used on the SmallWorlds graph. The top 12 recommendations from the *initial layout* were used. Initial layout was computed using the computational model described by the equations in the previous section, which use only profile overlap to generate recommendations. Importantly, we note that while MovieLens operates on a dataset of 10K items with 1M scaled preference ratings, SmallWorlds data is limited to the items listed in a users friends' Facebook profiles, which is relatively small and has potentially unlimited diversity. For example, entries such as "My Wedding Video". The key factor behind the good performance of SmallWorlds is that items have been pre-filtered based on pre-existing social connections on Facebook.

For the SmallWorlds run, the number of items in each user's graph varied depending on the movies they had listed and the number of friends they had on Facebook. On average there were 179 items in the graph, ranging from 25 to 521 (median 128). As discussed in the previous section, profile size ranged from 10 to 25 with an average size of 14 (median 15). The top 12 recommendations from each system were analysed as this is the default recommendation set returned by MovieLens QuickPick. The frequency of occurrence of the removed item in the top 12 predictions was recorded for each. On average 1.94 items appeared in the top 12 list on SmallWorlds, while for MovieLens the average was 0.82. In reality, users are more interested in prediction order, rather than average numerical ratings. To assess the performance of SmallWorlds on this top-n metric, relative positioning of items was recorded. For 9 of 17 users SmallWorlds had more movies in top 12 than MovieLens did, for 4 users MovieLens did better, and ranking position was tied for 4 users.

A similar analysis was carried out for number of items in top 5 and in first position (which would obviously be the ideal spot for the removed item). The average number of items in top 5 was 1.00 for SmallWorlds and 0.65 for MovieLens. For 7 of our users SmallWorlds got more items in top 5 than MovieLens did, but for 3 of our users MovieLens did better. For 7 users the predicted rank was tied. The average number of items in the number one spot for SmallWorlds was 0.47 and 0.18 for MovieLens, meaning that most of the time none of the removed items appeared in the number one spot. For 5 of our users SmallWorlds predicted more items in the number one spot than MovieLens did, while MovieLens only performed better in 1 case. For 11 users, both systems predicted an equal number of items in the number one spot.

Lastly, a winner-loser analysis was performed: all items listed in the top 12 for at least one of the systems were examined for each user. The number of times each system had the removed item ranked higher than the other was recorded. On average SmallWorlds had 1.88 items ranked higher than MovieLens did, while MovieLens did better for 0.65 items on average. In terms of number of users, SmallWorlds did better for 9 users while MovieLens did better for 4 and the systems did equally well for 4 users. Again, the overall performance of SmallWorlds was surprisingly high compared to the benchmark on the top-n metric, indicating that use of social connections can greatly boost the predictive accuracy of a collaborative filtering recommender system.

5. Conclusions

In this paper we have presented SmallWorlds, a live Facebook application with a visual interactive interface that can be used to control item predictions based on underlying data from the Facebook API. We have presented a novel layout and interaction algorithm which can be used for producing

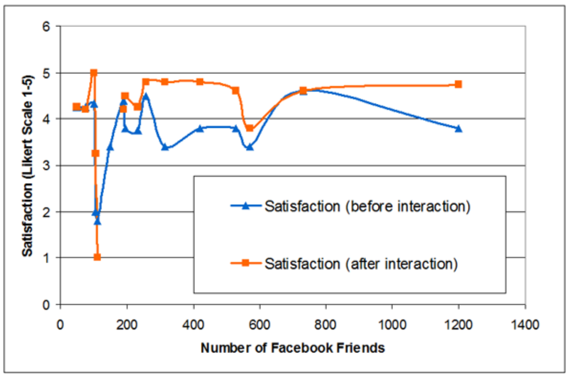

Figure 6: *Relation between satisfaction with recommendations and number of Facebook friends.*

item recommendations on a broad range of data. Results of a user study have been discussed which provide insight into the four research questions posed at the outset of this work. On the visualization side, our findings indicate that a visual interface does increase satisfaction in a system that makes predictions for users, because it incorporates transparency. Our results show that facilitating interactions with a visualization of a computational process provides a user with a sense of control over the outcome of that process, and therefore increases overall trust in the final result. On the social computing / AI side, our analysis has shown pre-existing friend connections can be used to boost satisfaction in predictions made using small datasets such as the data accessible from a single user on Facebook.

References

[BM98] BATAGELJ V., MRVAR A.: Pajek - program for large network analysis. *Connections 21* (1998), 47–57. 3

[BMZB05] BURKE R., MOBASHER B., ZABICKI R., BHAUMIK R.: Identifying attack models for secure recommendation. In *Beyond Personalisation Workshop at the International Confrece on Intelligent User Interfaces* (San Deigo, USA., 2005), ACM Press, pp. 347–361. 3

[Cyt] CYTOSCAPE: *Cytoscape Bioinformatics Appilcation*. Proprietary online application (nsf/nih funded), Cytoscape Consortium. available at http://www.cytoscape.org/. 3

[DVWK08] DANIS C. M., VIEGAS F. B., WATTENBERG M., KRISS J.: Your place or mine?: visualization as a community component. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), ACM, pp. 275–284. 3

[Dwy09] DWYER T.: Scalable, versatile and simple constrained graph layout. *Comput. Graph. Forum 28*, 3 (2009), 991–998. 6

[Fac] FACEBOOK: *Facebook growth statistics for 2009*. Press release, San Jose, CA. Jul 2007. available at http://www.facebook.com/press/info.php?statistics. 2, 4

[GBOH09] GRETARSSON B., BOSTANDJIEV S., O'DONOVAN J., HÖLLERER T.: Wigis: A scalable framework for web-based interactive graph visualizations. In *GD'09: Proceedings of the International Symposium on Graph Drawing* (2009). 3, 7

[GRGP01] GOLDBERG K., ROEDER T., GUPTA D., PERKINS C.: Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval 4*, 2 (2001), 133–151. 3

[Gro09] GROUPLENS: Movielens quickpick recommender system. `http://www.movielens.org/quickpick`, 2009. 10

[HB05] HEER J., BOYD D.: Vizster: Visualizing online social networks. In *INFOVIS '05: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization* (Washington, DC, USA, 2005), IEEE Computer Society, p. 5. 3

[HKR00] HERLOCKER J. L., KONSTAN J. A., RIEDL J.: Explaining collaborative filtering recommendations. In *Proceedings of ACM CSCW'00 Conference on Computer-Supported Cooperative Work* (2000), pp. 241–250. 2, 3, 9

[Kar01] KARYPIS G.: Evaluation of item-based top-n recommendation algorithms. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA* (2001), pp. 247–254. 3

[KPLB07] KANG H., PLAISANT C., LEE B., BEDERSON B. B.: Netlens: iterative exploration of content-actor network data. *Information Visualization 6*, 1 (2007), 18–31.

[LPP*06] LEE B., PARR C. S., PLAISANT C., BEDERSON B. B., VEKSLER V. D., GRAY W. D., KOTFILA C.: Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (November 2006), 1414–1426. 3

[MAL*03] MILLER B. N., ALBERT I., LAM S. K., KONSTAN J. A., RIEDL J.: Movielens unplugged: experiences with an occasionally connected recommender system. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces* (New York, NY, USA, 2003), ACM Press, pp. 263–266. 7, 9

[Mid02] MIDDLETON S. E.: Exploiting synergy between ontologies and recommender systems, semantic web workshop 2002 hawaii, usa, 2002. 3

[Mil67] MILGRAM S.: The small world problem. *Psychology Today*, 2 (1967), 60–67. 2, 3

[MKK06] MCNEE S. M., KAPOOR N., KONSTAN J. A.: Don't look stupid: avoiding pitfalls when recommending research papers. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work* (New York, NY, USA, 2006), ACM Press, pp. 171–180. 3

[MMN02] MELVILLE P., MOONEY R., NAGARAJAN R.: Content-boosted collaborative filtering. In *In Proceedings of the Eighteenth National Conference on Artificial Intelligence* (2002). 3

[OSG*08] O'DONOVAN J., SMYTH B., GRETARSSON B., BOSTANDJIEV S., HÖLLERER T.: Peerchooser: visual interactive recommendation. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (New York, NY, USA, 2008), ACM, pp. 1085–1088. 4

[OWS02] O'SULLIVAN D., WILSON D. C., SMYTH B.: Improving case-based recommendation: A collaborative filtering approach. In *Proceedings of the Sixth European Conference on Case Based Reasoning.* (2002), pp. LNAI 2416, p. 278 ff. 3

[RIS*94] RESNICK P., IACOVOU N., SUCHAK M., BERGSTROM P., RIEDL J.: Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work* (1994), pp. 175–186. 2, 3, 6, 9

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations, 1996. 3, 7

[SKKR01] SARWAR B. M., KARYPIS G., KONSTAN J. A., REIDL J.: Item-based collaborative filtering recommendation algorithms. In *World Wide Web* (2001), pp. 285–295. 3

[SM95] SHARDANAND U., MAES P.: Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems* (1995), vol. 1 of *Papers: Using the Information of Others*, pp. 210–217. 3

[SMO*03] SHANNON P., MARKIEL A., OZIER O., BALIGA N. S., WANG J. T., RAMAGE D., AMIN N., SCHWIKOWSKI B., IDEKER T.: Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res 13*, 11 (November 2003), 2498–2504. 3

[Sof09] SOFTWARE T. S.: Tom sawyer visualization, 2009. 3

[Tou] TOUCHGRAPH: *Touchgraph Navigator*. Proprietary online application, Touchgraph inc. available at http://www.touchgraph.com. 3

[War04] WARE C.: *Information Visualization: Perception for Design*, second ed. Morgan Kaufmann Publishers, 2004. 2

[Wat06] WATTENBERG M.: Visual exploration of multivariate graphs. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems* (New York, NY, USA, 2006), ACM, pp. 811–819. 3

[WBS*09] WILSON C., BOE B., SALA A., PUTTASWAMY K. P., ZHAO B. Y.: User interactions in social networks and their implications. In *Proceedings of the Fourth ACM European conference on Computer Systems (EuroSys)* (New York, NY, USA, 2009), ACM Press, pp. 205–218. 4

[ZJP08] ZHANG J., JONES N., PU P.: A visual interface for critiquing-based recommender systems. In *ACM Conference on Electronic Commerce* (2008), pp. 230–239. 3