Visible Relations in Online Communities: Modeling and Using Social Networks

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Andrew Webster

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

The Internet represents a unique opportunity for people to interact with each other across time and space, and online communities have existed long before the Internet's solidification in everyday living. There are two inherent challenges that online communities continue to contend with: motivating participation and organizing information. An online community's success or failure rests on the content generated by its users. Specifically, users need to continually participate by contributing new content and organizing existing content for others to be attracted and retained. I propose both participation and organization can be enhanced if users have an explicit awareness of the implicit social network which results from their online interactions. My approach makes this normally "hidden" social network visible and shows users that these intangible relations have an impact on satisfying their information needs and vice versa. That is, users can more readily situate their information needs within social processes, understanding that the value of information they receive and give is influenced and has influence on the mostly incidental relations they have formed with others. First, I describe how to model a social network within an online discussion forum and visualize the subsequent relationships in a way that motivates participation. Second, I show that social networks can also be modeled to generate recommendations of information items and that, through an interactive visualization, users can make direct adjustments to the model in order to improve their personal recommendations. I conclude that these modeling and visualization techniques are beneficial to online communities as their social capital is enhanced by "weaving" users more tightly together.

Acknowledgements

Having the confidence and support of Julita Vassileva, my supervisor, for the past two years is an experience that is hard to condense into a few understated sentences of gratitude. I was fixed on entering industry after attaining my B.Sc. because it was always the next step in The Plan and my course was set. However, due to the advice and encouragement from Dr. Vassileva, I discovered that The Plan was faulty, misconceived, and meant for someone else. Of course, I have never looked back since.

I must elaborate on the one thing I am most grateful for: the freedom I was allowed in my studies. Not all supervisors, who have numerous projects to follow and responsibilities to manage, are willing to let a Master's student constantly throw new things into the mix. Consequently, I must thank Dr. Vassileva for her support and guidance in exploring my own ideas while fostering the confidence that is often needed to see them through.

My family also requires recognition. Whenever I was feeling worn out, unsure, or discouraged, I could always go home and revitalize my spirit in their unconditional praise and support. And even if I didn't always express it, they knew I was appreciative.

Finally, I would like to thank my advising committee, Gord McCalla and Chris Zhang, for their comments and insights on this manuscript.

Contents

List of Tables

List of Figures

Chapter 1
Introduction

The Internet represents a unique opportunity for people to interact with each other across time and space, and *online communities* have existed long before the Internet's solidification in everyday living (see Rheingold, 1993). Since there is no widely accepted definition, I have chosen to broadly define an online community as a virtual social space where people meet and interact (Preece, 2001). Other indicators include (but are not limited to) shared interests, resources, goals, and identity. A critical aspect from the perspective of this research is that the success or failure of an online community is mainly dependent on its users to generate and sustain its purpose for existence. For example, a support community focused on a specific type of knee injury needs its users to continually contribute and clarify their experience with the injury in order for the community to retain its relevancy (c.f. Maloney-Krichmar & Preece, 2005).

Information, in general, can then be thought of as the lifeblood of online communities, and the interactions between users as the links through which it flows. The community fades if the information is stagnant or overwhelming; likewise, if the organization of information is disordered or constraining. The work presented here addresses these two challenges: motivating participation and organizing information. The former involves ensuring that new information is continually "pumped" into the community. The latter involves ensuring that users are getting meaningful information and experiencing meaningful interactions.

The overall hypothesis is that both participation and organization can be enhanced if community users have an explicit awareness of the implicit social network of interpersonal relationships which stems from their online interactions (between information items and each other). Therefore, my research enhances the *social capital* inherently present within online communities through technological means. Social capital is defined as the "investment in social relations by individuals through which they gain access to embedded resources to enhance expected returns of instrumental or expressive actions" (Lin, 1999, p. 39). Instrumental actions are actions that lead to new resources; expressive actions are ones that maintain existing resources. Investment in social relations is usually not strictly necessary for accessing resources within online communities since online communities are typically *open*, i.e. information items are available to anyone. It is usually detrimental to the community when a registration process (or similar policy) blocks access to information. Since anyone can access the community's information resources, social relations between users are not emphasized as a means to obtain or maintain resources. The exception is popular *social networking* web sites such as MySpace[1] or Facebook[2] that require users to build explicit "friendships" in order to gain access to certain types of information. Often these systems are primarily used to communicate and coordinate with real-life friends, colleagues, family members, etc.; the value of information is tightly constrained to individuals' immediate social network (e.g.: Who is going to the movie tonight? Has anyone heard of this new rock band?).

In communities that are built around the sharing of information/content with everyone, such as the photo sharing web site Flickr[3], social relations are not necessary for accessing the bulk of the community's resources and thus social capital is less tangible. These types of community are vulnerable to being "plundered" by newer communities/systems that have more novel features or are more accommodating of user demands, e.g. the ability to publish playlists of favourite songs. Consequently, I focus on information-sharing online communities, and my approach makes this normally "hidden" social network visible and shows users that these intangible relations have an impact on satisfying their information needs and vice versa. That is, users can more readily situate their information needs within social processes, understand-

[1]http://www.myspace.com
[2]http://www.facebook.com
[3]http://www.flickr.com

ing that the value of information they receive and give is influenced and has influence on the (mostly) incidental relations they have formed with others. Thus, the community becomes more “sticky”: it becomes harder for users to move to a different community since they would have to recreate their social networks at least partially (Bush & Tiwana, 2005). There are two components required to achieve this effect:

1. A way to discover and model implicit relationships between users; and,

2. A way to express the inferred relationships back to the users.

I will briefly review existing techniques related to each component. I begin with strategies and applications of modeling relationships in online communities from a mainly graph-based perspective (Section 1.1) before moving to social visualizations (Section 1.2) as a method to communicate back relationships.

1.1 Modeling Relationships and Online Communities

The process of discovering and analyzing community structures is multi-disciplinary and extends back to the social and behavioural sciences. Social network analysis (SNA) is a methodology that charts the ties between social entities (e.g. people, groups, companies, etc.) and analyzes the underlying graph structure to better interpret or predict the entities’ behaviour (Wasserman & Faust, 1994). For instance, SNA was used by Burkhardt & Brass (1990) to investigate what impact new technology has on the organizational structure within companies. Approximately 80 employees in a federal agency were given questionnaires asking them to identify the individuals whom they communicated with during a typical work week. The questionnaires were administered before, during and after the introduction of a new distributed computer system (most employees reported having limited computer experience). The differences in the resulting social networks showed that early adopters were able to reduce uncertainty for others and thus gained the ability to increase their power and *centrality* within the agency. Centrality is a network measure and asserts that nodes with more network ties are usually at an advantage over ones with fewer ties. Another example of SNA is the work of Espinoza (1999) that showed how the poor in the Peru were able secure scarce resources, such as jobs, during the economic turmoil of the 1990s.

In the previous examples, social networks were constructed after laborious ethnographic study (e.g. questionnaires, interviews, etc.). In an online setting, there is usually a large amount of rich, existing data to draw on. E-mail exchanges, discussion posts, and co authorship on scientific papers are excellent sources for inferring *interaction relationships* between users while browsing logs/history, search queries, and item ratings help infer *similarity relationships*[4]. Social networks built from interaction relationships are often the most straightforward to assemble. It is easy to detect if Person A responds to Person B's posts in a discussion forum more than N times (and vice versa) which is the basis for a new tie in the network. It is more challenging to contextualize that relationship without further knowledge of Person A, B and the situation in general (e.g. is A giving B repeated assistance?). However, the graph structure can reveal insights into how the community functions. The hierarchical divisive clustering algorithm in Tyler et al. (2005) automatically detects cohesive *subgroups* present within larger *components*. A subgroup is a tightly woven cluster of nodes that are loosely coupled to the remaining network, and a component is simply a set of nodes that are connected together. That is, the algorithm can determine whether the community consists of a single core group of users or if it is fractionalized into smaller interaction groups. This knowledge is of potential value to community developers as it can guide the design of new communities or the implementation of policies in existing ones. In the former instance, since there are core users, it can be safely assumed there are also peripheral users. It would then be prudent, for example, to investigate whether peripheral users feel welcomed in the community and if something can be done to better meet their needs.

[4]Privacy is always a concern when dealing with the types of aforementioned data, but the issue falls outside the scope of this research.

A more widespread approach is to relate users by similar behaviour and taste rather than by interactions. *Stereotyping* is a technique that is of interest to online communities as it predicts user behaviour. A stereotype is a set of attributes and characteristics that describes a set of users. These characteristics may range from demographic similarities (e.g. age, sex, education level, etc.) to similar navigation styles. Stereotypes can be constructed by hand or inferred from existing data by machine learning techniques like decision trees (Paliouras et al., 1999) or a combination of k-nearest neighbour, naive Bayes nets and weighted feature vectors (Lock & Kudenko, 2006). If a user can be accurately assigned to a stereotype, then there are opportunities to automatically support her needs (e.g. suggest new areas of the community to visit). The limitation is that sensitive, personal information must be provided by the user who may not always be willing to do so. The standout similarity technique, however, is *collaborative filtering* which correlates users based on the similarity of their ratings on items[5]. These correlations are exploited to predict a user's preference towards non-rated items. The main application of collaborative filtering is in information filtering and retrieval, namely *recommender systems* (Resnick & Varian, 1997). Recommender systems are prevalent in the e-commerce domain and typically do not feature in online communities. However, recommender systems usually serve a large number of people, and the implicit social network of similarity relationships can be structurally analyzed, as previously mentioned, to identify subgroups or (in this case) *communities of interest* (e.g. all users who like a certain selection of cult classics in a movie recommender).

The potential for recommender systems to support the development of online communities has been acknowledged (Terveen & Hill, 2001) but I am not aware of any specific research on this matter. The closest related system is an early one: Referral Web (Kautz et al., 1997) mined web documents for co-occurences of people's names in close proximity. When two names are discovered, an edge is formed in a social network and the document indicates what interests are shared between the pair. The intention was to make the hidden social network explicit to users as a nav-

[5]see Section 3.2 for a full discussion on collaborative filtering.

igational tool. For example, users could ask who was an expert in "computational complexity" and then find the shortest chain of referrals to one of the resulting experts. Referral Web is a means to satisfy individual information-seeking behaviour and is not intended to develop or manage social processes. However, there is recent interest in modeling social processes from a marketing and communications perspective. This approach is concerned with modeling *influence relationships* between people, i.e. how much the decisions of one person affect the decisions of another. The goal is to identify the smallest set of influential individuals within a social network whose adoption of a product would eventually trigger the maximum number of others in the network to adopt as well (Domingos & Richardson, 2001; Kempe et al., 2003). Those in the "trend-setter" selection could then be targeted with an appropriate *market action* such as a free product sample or discount. Influential users in an online community could also be singled out for special treatment. For example, they could be asked to beta test new features that community developers are thinking of implementing. Their response would be a good indicator of how the remainder of the community would accept and use the new feature.

1.2 Social Visualizations

This section explores the use of visualizations within online spaces as a "window" into the community of users. The seminal work in this area is focused on *social visualizations* (Erickson et al., 1999; Erickson, 2003). According to Erickson (2003) a social visualization is "a visual (or sonic or other perceptual) representation of information from which the presence, activities, and other characteristics of members of a social collectivity may be inferred, and, by extension, can provide the basis for making inferences about the activities and characteristics of the group as a whole" (p. 846). The central concept is that people are better able to align their interactions when social cues are available. An example cue is that we may delay engaging someone in conversation when we see that she is on the telephone, etc. Such cues are inherently lacking in online environments. The Babble system is a chat application that visualizes each participant as a small coloured dot at the edge of the visualization window (Erickson et al., 1999). As participants use Babble (e.g. typing or even scrolling the chat window), their respective dot begins to move towards the center of the visualization. The dots drifts back to the edge over a 20 minute period if participants are inactive. The purpose is to create an awareness of who is present in the chat and their current level of involvement/activity. PeopleGarden is a similar type of visualization where each user is represented as a "flower" that grows and blooms depending on the length of time a user has been in a discussion forum and the number of posts she contributes, respectively (Donath, 2002). Each PeopleGarden "snapshot" characterizes the activity style of each user (e.g. users with small, short flowers have just joined the conversation) and the style of each conversation (e.g. a handful of users may be dominating). The intention is to prompt reflection that the online space is a social space and should be approached as such (e.g. following etiquette such as letting others have their say). Previous work from my lab has also developed visualizations that prompt user reflection. In Sun & Vassileva (2006), a user is represented as a single star in the night sky. The size and brightness/colour of the star is dependent on the number and quality of the particular user's contributions, respectively. Users engage in a form of *social comparison* when comparing their star to others and have an incentive to contribute more to the community in order to become the biggest and brightest star in the sky. Users are not the only ones who can benefit from reflecting on social visualizations: community designers/managers (or similar role) are another target audience. Brooks et al. (2006) employed a *sociogram* to help teachers get an impression of student activity within long-distance learning courses, specifically the pattern of replies in a discussion forum. Learners were represented as nodes positioned along a ring. If one learner had replied to another's post, then the two were linked together. And the more posts a learner contributed, the larger her node was. It was then a straightforward task for teachers to make casual observations of who was participating, who they were participating with, and by how much.

While the previous visualizations conceptualize the activity of users, they do not propose any explicit link or relationship between users. This is intentional as Erickson (2003) suggests that it is difficult to anticipate every use of the system, so users are better equipped at interpreting the visualizations themselves. He also warns that built-in interpretation is a poor addition because social visualizations should be able to "deceive" because humans are adept at providing misleading cues (e.g. feigning interest) when appropriate. I suggest that linking users together in a visualization is an acceptable form of system interpretation provided that the link is still open to user interpretation. Visualizing the community's activity is a good first step; however, the next is to contextualize the current user's activity at a *person-to-person* level within the wider community. What we do and how we do it is strongly motivated by our relationships with others, especially when it comes to finding and receiving information, and ContactMap is an application that organizes communication information by visually representing the user's personal social network (Nardi et al., 2004). Each person the user has contact with is represented by a photograph of that person and a label. Each contact is clustered together with similar contacts (e.g. colleagues in Human Resources) and given a representative colour. Those contacts that the user communicates the most with appear in the center of the screen while those with less frequent interaction appear at the edge. By selecting a contact, a user can view all the documents she has exchanged with that contact over e-mail, etc. While not directly related to online communities, this example highlights a direction that online communities can take by revealing relationships between community users.

1.3 Outline

The challenge of motivating participation revolves around attracting and retaining a *critical mass* of users. The odds that a community can attract new users are significantly improved if there are already a number of users who contribute to the community. Thus, there is no firm number of users which defines the threshold for critical mass. Instead, it depends on the number of actively contributing users versus the number of passive users or so-called *lurkers*. Generally, this ratio is heavily skewed towards lurkers and online communities can expect 45-90% of their users not to contribute (Nonnecke & Preece, 2000). Thus, effective incentives or techniques that motivate participation are of considerable value. In Chapter 2, I demonstrate an approach that "weaves" lurkers into a discussion community by creating a generalized awareness of the relationships they form with active participants through reading and browsing of posts. Specifically, relationships are represented by a social visualization that is embedded within each and every discussion post. Once aware of these relationships, lurkers are more inclined to reciprocate with their own contributions. Conversely, active participants can determine who makes up their (invisible/silent) audience and engage individual audience members, if so desired.

Chapter 3 takes a different approach: it models the similarity relationships between users and shows that the subsequent social network can be used to effectively distribute and recommend information items, i.e. the word of mouth process. There has been recent interest in how social processes, like word of mouth, can be exploited to better satisfy information-seeking goals (c.f. Perugini et al., 2004). The advantage is that social processes (i.e. the relationships/connections formed between people) are self-organizing–a property which can be potentially "unlocked" by an automated system. Chapter 4 examines a news recommender system, KeepUP, that was implemented using the work done in Chapter 3. KeepUP explores strategies in supporting "communities within communities" which develop from the distribution and recommendation of information items within the social network.

The main strategy is a visualization that reveals to a user who her *neighbours* are and how much *influence* each neighbour has on her recommendations. The visualization is interactive and allows users to change the influence from their neighbours, giving users some control over the recommendation process.

Chapter 2
Motivating Participation

The types of interactions within online communities are diverse and may include exchanging information or social support (Maloney-Krichmar & Preece, 2005), fostering social ties (Boyd, 2004), supporting learning (Johnson, 2001), extending real-world relationships/communities (Wellman et al., 1996), or a combination of these. The crux of building online communities is on successfully entangling people together around a common purpose (that is usually reflected in the developer's agenda)[1].

It is a well-known dilemma that a certain amount of interaction/contribution must occur in an online community before users start perceiving the benefits of the system and become active participants themselves. This problem is especially acute and frustrating for developers who must reconcile that the majority of their membership (45-90%) never participates (Nonnecke & Preece, 2000) within systems understood to be gift economies (Rheingold, 1993; Smith & Kollock, 1999). In a gift economy, information is exchanged for the benefit of the whole community with the generalized understanding that the contributing individuals will receive some benefit from others later on. Hidden, non-participating users (lurkers) do not reciprocate the benefits they have received and have been generally seen as destructive to the health of online communities (Smith & Kollock, 1999). However, more recent research (Preece et al., 2004; Takahashi et al., 2003) has shown that this is not the case. Lurkers were interviewed and reported feeling a sense of belonging to the community even though they had lower satisfaction with the community than participating

[1]This chapter contains sections of Webster & Vassileva (2006) that are reprinted here with the kind permission of Springer Science and Business Media.

users. The interviewed participating users viewed lurkers as legitimate users of the community (akin to the importance of having an audience in theater performances). It was suggested that "lurking should be recognized as a bona fide activity and supported more effectively" (Preece et al., 2004, p. 216).

However, in the early stages of a developing online community, motivating participation in everyone is crucial. Previous research from my lab (Cheng & Vassileva, 2005) suggested awarding or revoking social status based on contribution levels would increase contribution because people would be motivated by social comparison (i.e. their status in the community) and would fear losing their current standing within the community if they did not continue participating. Unfortunately, rewarding contribution with social status (or any other type of explicit reward) does have a significant pitfall. There are numerous studies in psychology that suggest intrinsic motivation to complete a task is negatively affected when an extrinsic reward is introduced (Deci et al., 1999). For example, children who draw pictures with coloured pencils are more likely to switch to regular pencils after being given a certificate of achievement for using the coloured ones. There is a high probability that active participants choose to contribute to a community because they feel it is already a worthwhile endeavour, and the introduction of extrinsic incentives degrades the intrinsic motivation. Therefore, I propose a more subtle mechanism to motivate participation by making the implicit social network of interpersonal relationships between community users explicit and visible through a visualization. The principal aim is to connect lurkers to active participants, "weaving" them into the community.

2.1 Related Work

The question of what motivates or triggers individuals to join and participate in online communities and how to design the technical features of the community software accordingly rests on the particular rationale from a wide range of perspectives. Preece & Maloney-Krichmar (2003) identify research in social psychology, sociology, communication studies, computer-supported cooperative work (CSCW) and human-computer interaction (HCI) as main areas which can help inform designers about how and why people interact in online communities. Consequently, there are many guiding directions on which interactions to support and how to support them. The variety of online communities with their own specific sets of interactions (e.g. a mailing list for cancer-sufferers vs. an interactive, educational website for teens) and specific purposes makes it very hard to choose appropriate guidelines for interaction design.

An area dealing with social issues in interaction design is CSCW and its application of theories from social psychology to the problems of group work (Grudin, 1994). Collective effort, social identity, and social categorization (Hogg & Tindale, 2001) are all theories which have provided direction in the design and evaluation of technical features to support the work of groups (Kraut, 2003). These theories have also been used in the design and study of online communities (Beenen et al., 2004; Dholakia et al., 2004). However, there is no unified theory in social psychology and most theories are "mid-level," i.e. only the behaviour of individuals within groups is explained. Also, online groups have only recently received attention from social psychologists, and it is not completely clear what similarities and differences exist between face-to-face and online groups (Hogg & Tindale, 2001). Finally, the CSCW agenda is one of supporting groups that primarily exist to achieve specific work-related goals (relatively short term, requiring close collaboration by the group users). Therefore, not all online communities can take straightforward advantage of these fields of knowledge, especially those that are interest-driven rather than goal-driven. For example, in investigating whether the theory of collective effort could potentially aid in increasing participation (the number of movie ratings) in the interest-based MovieLens community, Beenen et al. (2004) did observe an increase in the number of contributed ratings but failed to attribute it directly to the implementation of the theory. The authors offer several reasons for this including "a deeper mismatch of goals and values of HCI and CSCW research with those of social psychology" (p 220).

I suggest that the failure to apply the theory may also be due to the tendency to link non-participation with free-riding. This is a connection which is hard to avoid within a collaborative work context where individuals must work on their tasks to be of value to the group or community. Therefore, from the perspective of CSCW, non-participants are treated as a problem to be fixed. However, in a community where people share common interest but not a task or goal, lurking is acceptable.

It is often difficult for new users to join a new or preexisting community. It takes time to uncover the structure, norms, and history of the community before making one's presence known. It would be useful to present the rudimentary relationship that lurkers form with others, even when they are simply reading or browsing information. My hypothesis is that by making the structure of these relations explicit new communities will develop quicker by rapidly integrating newcomers, increasing the probability that they will become active contributors rather than remaining on the sidelines as lurkers. In the next section, a mechanism for modeling such relations is described.

2.2 Mechanism: Energy and Relations

I place value on the act of contributing in online communities and not just necessarily on what is contributed: information valuable today may be worthless tomorrow. It is important to have people who are invested in each other enough to share information, exchange support, etc.

2.2.1 Energy: The Building Block

First, I introduce the concept of *energy* in an online community which is a measure of the current level of contributions in the community. When an item (e.g. discussion post, movie review, blog entry, etc.) is contributed, it brings in a default number of new *energy units* into the system. For example, a new post in a discussion thread may produce 5 units.

Only a certain number of energy units are allowed to stay attached to the new contribution (e.g. by default a post may keep 3 of the 5 units). The number of these

Figure 2.1: The visual appearance of contributions at different levels of energy.

units determines the contribution's *visibility* in the community. Different levels of visibility are achieved through the scaled use of colour and font size. If a contribution possesses many units, then it will be rendered with hot colours (e.g. orange, yellow) and large fonts, advancing towards the viewer. Conversely, if an item has few or no units, then it will be rendered with cold colours (e.g. blue, purple), and small fonts, receding from the viewer (see Figure 2.1).

Units kept by an item are considered to be in the *@work* state (i.e. the energy units work to make the item more visible) while units not kept are considered to be in the *stored* state, i.e. a communal pool of units that are available to *all* users and can be moved into the @work state. Energy units can freely move between the stored and @work states; this movement is mainly dependent on the actions of the community's users. If a user positively evaluates an item (and stored energy is available) then she may decide to "heat it up" by moving a stored energy unit into that item (equivalent to rating the contribution). As a result, the item becomes a little more visible to all other users. Conversely, other users may negatively evaluate the same item and "cool it down" by moving energy units back into storage, one at a time. There are 4 simple rules governing how energy may be distributed:

1. A user cannot heat up (i.e. add energy to) or cool down (i.e. remove energy from) items she has contributed.

2. A user can only heat up and cool down an item once.

3. Items can only be heated up if stored energy is available.

4. There is a set upper limit on the number of energy units an item may hold.

Community users should not be able to add energy to their own contributions as their judgment is biased (rule 1). It should not be possible for one user to have too much influence over the visibility of a particular contribution, i.e. each user has one vote per item (rule 2). Energy can be added to contributions only if there is stored energy in the community, i.e. the community must manage the shared, limited resource of what is and is not visible at any point in time (rule 3). The concept of community energy provides a novel metaphor and system for rating content with a number of advantages:

1. Energy is finite and depends on the number of contributions to the community, keeping the ratings always in proportion with the contributions (i.e. prevents inflation in the ratings).

2. Using community energy units for evaluation encourages the user to reflect on the usefulness of the item to the community and not just to herself (i.e. "I want others to notice this item" or "I want others to ignore this item").

3. Evaluation is cognitively less demanding than determining if an item deserves 1, 2, 3, 4, or 5 stars, for instance. The user simply determines if the item should be more or less visible.

4. Emphasis is placed on the act of contributing (i.e. each contribution brings in new energy–a useful resource to the community).

The movement of energy immediately changes the “visual landscape” of the community, reflecting and highlighting users’ action within the system. I suggest this is especially relevant when items are rated and propose that the visual properties of items should immediately be effected. A short animation of 2 to 3 seconds duration could be shown that cycles the item’s background colour from either blue or orange to an intermediate colour depending on how the user rated (i.e. heated up or cooled down). For example, an item currently holding 7 energy units would have the visual appearance (e.g. background colour, etc.) as depicted in Figure 2.1, and if a user added an energy unit to that item, then she would see the item’s visual appearance cycle from the style of 0 energy units to the style of 8 energy units. From her perspective, it provides a fun, rewarding experience that impresses upon her the immediate effects of her actions within the community. From a developer’s perspective, it is meant to help users navigate the community, e.g. to find the most current and relevant items. The combination of hot colours and large font size were chosen to make “good” items instantly leap off the page while dark colours and small font size were chosen to make poor items easy to ignore. Therefore, the individual act of rating becomes a method of directing others away or towards particular items. There is also a possibility of *social comparison* when one user compares the “brilliance” of her contributions to those around it, and this may compel her to contribute more items in order to capture more energy.

The energy metaphor is a type of *social navigation* (c.f. Dieberger et al., 2000). The classic example of social navigation is “footprints in the snow.” Over time, the aggregated actions of users reveal a pattern: i.e., a path that is followed to outstanding items in the community. However, one problem with this approach

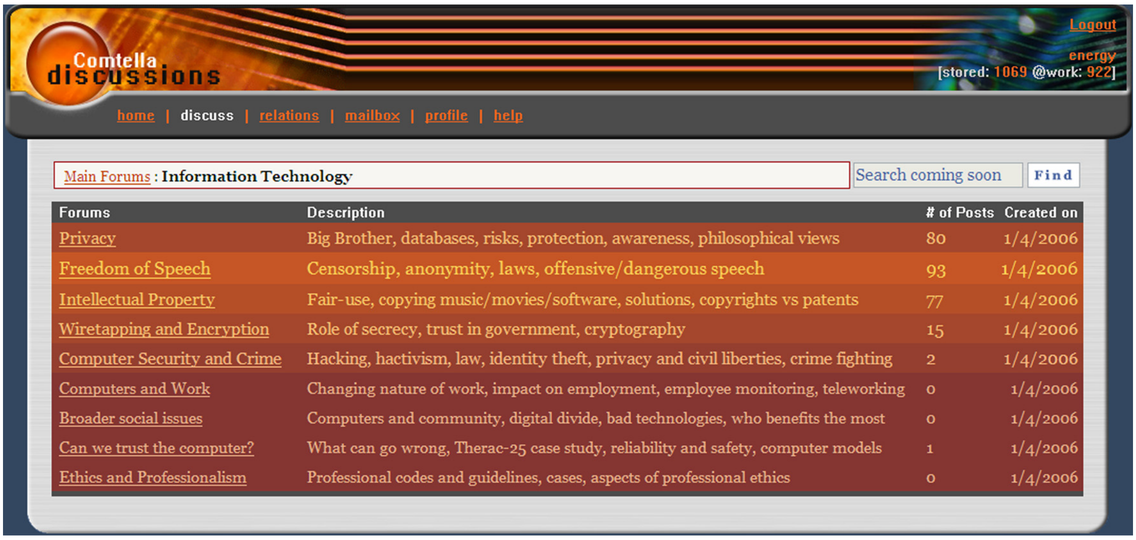


Figure 2.2: An example of the distribution of energy in an online discussion forum.

is that these paths have the tendency to lead to a few, popular items which will eventually grow stale. The novel feature of the energy metaphor is that a renewable resource (i.e. energy units) needs to be managed by the community in light of natural factors like the decay of energy out of the community or social factors like a person who is rating, perhaps, too aggressively. In combination, these features allow users to easily determine where activity in the community is occurring and what particular activities are relevant to the whole membership *at the present time* (e.g., see Figure 2.2). This should be of particular benefit to new users who are trying to decide what the community presently values in order to best introduce their contributions, opinions, values, etc.

2.2.2 Modeling Interpersonal Relations

Modeling and visualization for interpersonal relations aims at three goals: 1) connect lurkers and contributors, 2) give the viewer opportunity for reflection which can be beneficial, as suggested by open user modeling approaches (Bull et al., 2003), 3) influence the viewer to modify her behaviour in a desired way (to participate more). The visualization should also be dynamic to reflect that individual actions constantly modify relationships and in this way confirm and reward the user's actions.

The most common relationship found in online communities is the weakest (making it difficult to capture): the lurker-contributor relationship. The importance of weak ties has long been recognized (Granovetter, 1973) so defining a tenable connection between lurkers and contributors is a desirable feature of the visualization but also a challenge.

A relationship between two users A and B always has two sides: from A→B and from B→A, which are not necessarily symmetrical. I define the notion of *user visibility* to capture the inherent asymmetry in interpersonal relationships. User visibility is a value ranging from 1 (invisible / unknown / opaque) to 0 (completely visible / transparent). For example, when a new user enters the community, she does not know or "see" any other user. Thus, from this user's perspective, visibility values of 1 are assigned to all other users, i.e. her relationships with all other users of the community have value 1. Conversely, as she is a new user, all other community users will assign a value of 1 to their relationships with this new user. The assignment of 1 instead of 0 to mean "invisible" may seem counter-intuitive to many but was chosen to reflect "distance" between users. Thus, two users who do not know each other will be a greater distance apart compared to users who do know each other well (as will be shown in the visualization in the next section). However, exact visibility values are never listed to users, so it is more of an implementation decision on the developer's side whether to flip the meaning around.

The visibility value at one end of the relation pair is dependent on actions performed by the user on the other end (see Section 2.2.4). For example, if a lurker reads several messages in a discussion forum, then the authors of these messages will become slightly more visible to the lurker (i.e. the value of the lurker's relationships with the authors of the posts will decrease), yet the lurker's visibility for the other users still remains unaffected (i.e. their relationships with the lurker will still have value 1).

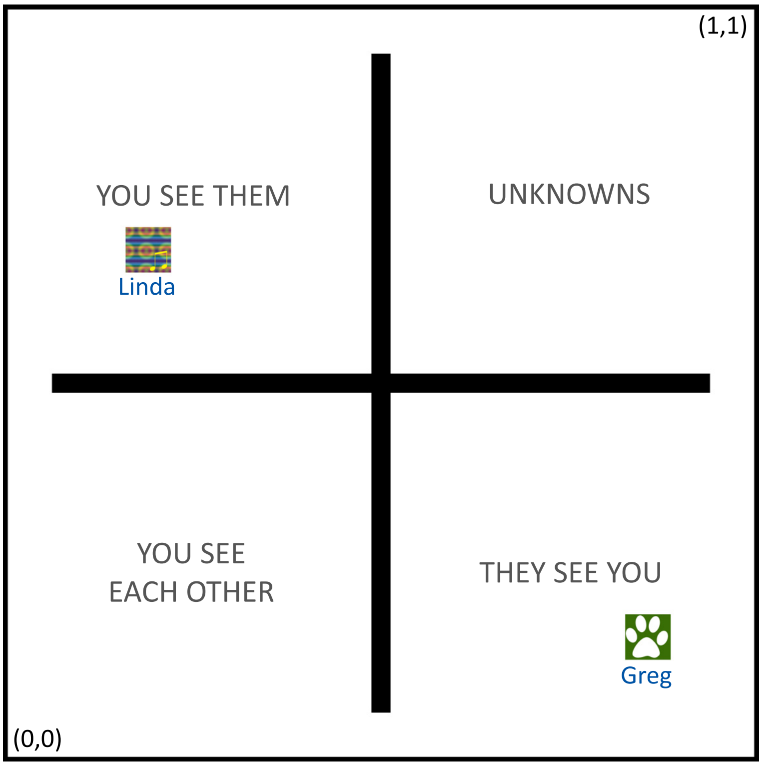


Figure 2.3: Example relation visualization (Relavis) from Ralph's viewpoint.

2.2.3 Relation Visualization (Relavis)

The relation between two individual users can be visualized in a two-dimensional space which I call a Relaviz (see Figure 2.3). The horizontal axis (0 to 1) indicates the visibility of other users to the visualization's viewer (in this example, Ralph) while the vertical axis (0 to 1) indicates the visibility of the viewer to the other users. For example, in Figure 2.3, the position of Linda's avatar icon describes the relation where Ralph frequently accesses content created by Linda but the reverse is not true.

To assist reading, the space is characterized by four relation quadrants: "you see them," "unknown," "you see each other," and "they see you." Insignificant relations (i.e. unknowns) are located in the top-right corner with coordinates (1, 1) while more significant relations (i.e. mutual awareness) are located in the bottom-left corner with coordinates (0, 0).

Let us return to the scenario where a lurker reads posts in a discussion forum.

Let Ralph be an active contributor, checking his Relaviz once in a while to see how things stand. This time he notices "Greg" in the "they see you" quadrant (who did not appear the last time Ralph checked). Ralph can guess that Greg has read and rated positively most, if not all, of Ralph's contributions since the relation is so strongly asymmetric. Depending on the size of the community, Ralph may guess that Greg is new in the community or a chronic lurker who has recently discovered his contributions. This discovery gives an opportunity for Ralph, who has already received some benefit (i.e. Greg adding energy units to Ralph's contributions), to directly communicate with Greg, to search for Greg's contributions and perhaps evaluate them.

If Greg looks at his Relaviz, logically, he will see Ralph appear in the "you see them" quadrant. The important consideration is that both users now have some awareness of each other and can take actions to further define the relation. In order to encourage the use of the Relavis, whenever possible, a light-weight version is displayed alongside the contribution to give specific relation information (see Figure 2.4).

2.2.4 Calculating Visibility Values

The calculation of visibility values is largely dependent on the features of the online community and relatively straightforward. Actions which are deemed to affect the visibility between users are assigned constant values which will either increase or decrease the overall visibility value (recall it ranges from 0, visible, to 1, invisible). In my implementation, accessing a discussion thread subtracts a little (-0.005) from the opaqueness of each reader-author relationship regardless whether the reader actually reads the specific post or not. Explicit actions that indicate preference (e.g. "heating" (-0.05) or "cooling" (+0.05) posts) have a stronger impact on visibility, and items' energy units come into play to provide bonuses: "hot" items have a stronger effect on changing visibility than "colder" ones. However, the determination of these constants is an open question. Some initial intuition is required to say certain actions affect visibility between two community users more than others. The analysis of the results the evaluation (described in the next section) should provide direction into how these values should be best determined.

2.3 Study

Comtella Discussions[2] (CD) was an online discussion forum that was implemented using the previously discussed energy and visibility metaphors. It was used in a study that tested the effectiveness of these features in motivating contributions within a community of university students. Access to content was restricted to registered users but anyone was able to create an account after consenting to the conditions of the study. A nickname (i.e. alias or pseudonym), e-mail address, and password were required to create an account, so students were free to be relatively anonymous and create multiple identities, if they desired.

2.3.1 Participant Groups

CD was used by students from two university courses at the University of Saskatchewan: Computer Science 408 and Philosophy 236, from January to April 2006. Both courses studied the ethics of technology except the former emphasized information technology while the latter emphasized ethical theory and biotechnology.

Table 2.1: Subject groups in Comtella Discussions.

Label	N	Description
C_α	10	Core users (computer science students) who are required to participate and who see the test interface.
C_β	9	Core users who see a control (standard discussion forum) interface.
P_α	15	Peripheral users (philosophy students and others) who are not required to participate and who see the test interface.
P_β	17	Peripheral users who see a control interface.

[2]http://fire.usask.ca

Figure 2.4: A post header as seen by an α-group participant (left) and β-group participant (right).

The computer science students, as part of their coursework, were required to submit five posts to the forum every week. Thus, they represent the core membership of the community. Conversely, philosophy students were not required to participate and will represent peripheral users: their instructor recommended CD as an additional class resource. I denote the core users with C and peripheral users with P, and I divided users (by the order in which accounts are created) into two orthogonal subgroups: a test group which saw the energy interface and Relaviz visualizations (α) and a control group which experienced a standard discussion forum interface with no relation visualization (β). A summary of the groups is shown in Table 2.1. All groups used the same concept of community energy to evaluate postings, but the representation of the act of rating was different between groups (see Figure 2.4). Visibility relations were computed as described in Section 2.2.4 between **all** participants; however, only the α-group participants saw the Relavis that depicted their relations. To the β-group participants, CD had the appearance and functionality typical of online discussion forums.

2.3.2 Hypothesis

The hypothesis is that the subgroups using the α-interface, i.e. the test-subgroup, in both the core and peripheral user groups will show higher participation, will have less lurkers (or the number of non-actively participating users of the P_α group will be less than the corresponding number in the P_β group) and will show increased satisfaction with the community. In order for the hypothesis to hold, participation rates p of each group should be ranked in the following order:

$$p(C_\alpha) > p(C_\beta) > p(P_\alpha) > p(P_\beta) \tag{2.1}$$

As a consequence, if the hypothesis holds, I expect the average interaction levels (and, of course, the corresponding mutual visibility values) between pairs of users of the four groups will be partially ordered so that the mutual visibility of users of the α-subgroup in both the core and the peripheral group is highest. Also I expect that the users of the β-subgroups will be more visible for the users of the α-subgroups than the reverse in both the core and peripheral groups. The lowest visibility and interaction levels will be between users of the β-subgroups in each of the core and peripheral groups.

2.4 Results

As shown in Table 2.2, for most participation metrics, the expected order (Equation 2.1) between the groups holds. However, the only observed statistically significant result was that P_α subjects logged into the system more than P_β subjects did ($p < 0.02$).

Table 2.2: Subject group participation data.

	Contribution Counts				Average Accesses/Views		
Group	Threads	Posts	Comments	Evaluations	Logins	Threads	Relavis
C_α	72	326	17	55	66.3	233.6	4
C_β	60	299	5	11	48.6	180.2	n/a
P_α	6	10	0	6	15.9	28.1	1.1
P_β	1	6	1	4	7.9	19.2	n/a

Table 2.3: Interaction between subject groups.

Grouping	Interaction (from $\rightarrow$ to)	# of Relations	Avg. Visibility
Core to Core	$C_\alpha \rightarrow C_\alpha$	89	0.5988
	$C_\alpha \rightarrow C_\beta$	90	0.5763
	$C_\beta \rightarrow C_\alpha$	88	0.6125
	$C_\beta \rightarrow C_\beta$	72	0.6573
Core to Periphery	$C_\alpha \rightarrow P_\alpha$	11	0.9784
	$C_\alpha \rightarrow P_\beta$	7	0.9860
	$C_\beta \rightarrow P_\alpha$	11	0.9894
	$C_\beta \rightarrow P_\beta$	3	0.9820
Periphery to Core	$P_\alpha \rightarrow C_\alpha$	82	0.9624
	$P_\alpha \rightarrow C_\beta$	87	0.9674
	$P_\beta \rightarrow C_\alpha$	70	0.9711
	$P_\beta \rightarrow C_\beta$	79	0.9742
Periphery to Periphery	$P_\alpha \rightarrow P_\alpha$	42	**0.9713**
	$P_\alpha \rightarrow P_\beta$	28	**0.9678**
	$P_\beta \rightarrow P_\alpha$	40	**0.9688**
	$P_\beta \rightarrow P_\beta$	33	**0.9667**

Table 2.3 shows the relative ordering of average visibility of the participants from each subgroup. For an idea of the level of interaction these average visibility values capture, consider if all incoming relations to a particular participant averaged a visibility value of 0.75, then this participant can expect that each user connected with an incoming relation to her has viewed at least one of her posts approximately 50 times (ignoring other actions such as heating and cooling).

The results generally conform to expectations. In particular, the P_α subjects interacted with the core group, C, more than P_β subjects did ($p < 0.01$) which was that basic objective. Within the core group, the users of the α-subgroup engaged in more symmetrical relationships. Eight (8) relations of mutual recognition (i.e. "you see each other") were made within the C_α group, compared to 3 such relations formed within the C_β group. The interactions and visibility among the users of the peripheral group, however, do not confirm my predictions. Even though the differences are small, the relationships of the P_β subjects among themselves and with P_α subjects show that they engaged in more interactions compared to P_α subjects (bold-face text in Table 2.3).

2.5 Summary

In this chapter, I proposed a new mechanism for motivating participation in interest-based online communities which engaged lurkers through modeling and visualizing the relations they build with other community users when reading, evaluating, commenting or replying to their contributions. The mechanism is based on ideas from open user modeling, a concept of community energy, and a new mechanism of rating contributions and visualizing the rank of contributions in the community interface. The results indicate that the new approach can draw increased participation for both active and non-active users. Fortunately, the computer science students reported that, generally, they liked the system and that it helped foster discussion among them. Unfortunately, students knew each other in real life from being present in the same course/classroom and much of the "core" discussion ended up occurring outside of Comtella Discussions.

The immediate, visual feedback provided after users rated a post was effective in motivating participation. From Table 2.2, it is clear that α users contributed significantly more ratings (i.e. "evaluations") compared to β users (61 to 15, respectively). While subtle, the visual "heating up" or "cooling down" effect that followed the user's act of rating was fun, fitted the energy metaphor, and didn't distract the users' attention or pose an additional cognitive load. The energy system acted as an implicit recommender function which is an important feature to have, especially in large communities. The next two chapters explore in depth how social networks can be used within a recommender system to help support online communities.

Chapter 3

Organizing Information

The energy and visibility metaphor of the previous chapter briefly touched on an interesting aspect of social navigation: self-organization. Due to the aggregated actions of users, a useful pattern develops which helps users find "good" items. Unfortunately, social navigation does not scale well. Consider that Flickr receives thousands of digital photos *per minute* and it is easy to imagine that some truly wonderful and exciting photographs will escape the attention of most community users because the social paths leading to these photographs will never "catch on." In large communities, like Flickr, the amount of available content is overwhelming, and having too much content/information is just as dangerous to an online community as having too little. Users become easily fatigued and overwhelmed, especially when looking for something specific.[1]

Recommender systems are a successful and widely popular solution to information overload (compared to social navigation techniques). Their goal is to find items of personal interest for individual users who would profit from timely and relevant recommendations. Recommender systems have done well in the e-commerce domain, and Amazon's[2] recommendation features have been widely noted (Linden, 2003). Yet, it is unusual to find even a large online community that makes use of a recommender system although there are many that use some form of social navigation. There may be several reasons for this: recommender systems are inherently complex, require continual management, and take time to be adapted to a particular

[1]This chapter contains sections of Webster & Vassileva (2007) that are reprinted here with the kind permission of Springer Science and Business Media.

[2]http://www.amazon.com

application domain (e.g. e-commerce). They may also be considered too inconsistent and underwhelming with their prediction accuracy to be of any real value. Also, it is often required that users explicitly rate items. If subsequent recommendations are poor, then the user is not properly compensated for her effort in rating items. Whatever the case, one thing is clear: recommendations do not simply “happen” out of users’ actions and must be coerced out of data using a variety of techniques, ranging from the statistical to the probabilistic. Even when using the most straightforward statistical technique, it is often difficult to explain to users the reason why an item was ultimately recommended.

There is the possibility that recommendations *can* just “happen” out of users’ actions if the system supports the right kind of actions that allow self-organization to occur. I base this on the observation that other information retrieval and filtering systems have successfully exploited *implicit recommendations*. For instance, linking to a web page can be thought of as recommendation of that page, and the well-known work of Kleinberg (1999) analyzes this link topology to identify *hubs* and *authorities*. This knowledge can be used to identify good sources of information for a certain topic, i.e. authorities, and Google’s[3] PageRank (Brin & Page, 1998) is an extensively modified version of this insight. Also, *collaborative tagging systems* (Golder & Huberman, 2006) demonstrate successful item classification by having users provide manual classification through a set of freely-chosen keywords, or *tags*, rather than relying on automated analysis or domain experts. A tag is viewed as the user’s “vote” for the item’s classification. When users search for items, they see the collective recommendation of what items are believed to match that query. Finally, even online communities can be thought of as implicit recommender systems. Users cannot always make personalized recommendations to individuals, but they can make suggestions to the general community as it can be safely assumed that common interests, goals, etc are shared between people in the community. For example, a user may spontaneously share her experience that one particular product is superior to another without being prompted by a specific question or request for

[3]http://www.google.com

information. If others felt that her reasons were not sufficiently justified or that their experience has differed, then they can always ask for more clarification or add their own opinion to the discussion.

Overall, recommender systems are distinctively "black box" systems (Herlocker et al., 2000): ugly and incomprehensible. From my perspective, I view them as hijacking an inherently social process, *word of mouth*, and placing themselves between people as authoritative intermediaries. It appears to the user that she is engaged in a dialog with the system–not her peers–about what to view next, although the system may be associating her with other like-minded users in order to predict items of interest. While conversational recommender systems (e.g. Burke et al., 1997) do engage users in an active role by means of a dialog, users are still left separately conversing with the system and not with their peers. This prompts the question of why can't recommender systems be more social? And I am not the first to ask this question (Perugini et al., 2004; Terveen & Hill, 2001). This chapter and the next examines a new direction in building recommender systems that closely follow how information is distributed in real life, i.e. word of mouth, something that early recommender systems were said to automatically replicate (Shardanand & Maes, 1995). Specifically, I rely on *information diffusion* models (Rogers, 2003, c.f.) in order to use a social network as the primary means in distributing and recommending information items. It is my goal to develop a system that allows for the self-organization of *communities of interest* and exploits this self-organization to make better recommendations that are more explainable, timely, and relevant. The focus of this chapter is on building and demonstrating an algorithmic framework which makes this type of system possible, and, as such, it is heavy on theory and light on application. However, chapter 4 applies the algorithms to a news recommender system, KeepUP, which includes a visualization that gives users insight into how the system eventually arrives at a recommendation.

3.1 Related Work

Users' activity within a recommender system has been acknowledged as "inducing an implicit social network and [influences] the connectivities in this network" (Mirza et al., 2003, p. 134) and that "recommendations are not delivered within a vacuum, but rather cast within an informal community of users and social context" (Perugini et al., 2004, p. 131). The former statement observes that similarity of interests, etc. is the basis for an implicit social network in recommender systems rather than the explicit and implicit interaction between users as was seen in chapter 2. The latter statement recognizes that the social context in which recommendations are made should not be discounted. Both authors make the argument that more attention needs to be placed on how social networks can be advantageously modeled and exploited to enhance users' experience (both with a recommender system and an online community). User modeling, either direct (e.g. using explicit input like item ratings) or indirect (e.g. data mining e-mail logs), and the computed similarity between user models was seen as the primary means to obtain social networks that are exploitable by the recommendation process either through structural analysis or by embedding additional information into connections between users. For an example of structural analysis, recommender systems (in general) were evaluated in light of the network structure created between users under certain conditions (Mirza et al., 2003). One condition that was analyzed was the minimum number of shared items users must rate in order to be connected all together. It is believed that knowing this number would help the system's developer strike a balance between ensuring good recommendations and not alienating users with too much work. For an example of the latter approach, explicit indication of trust between users was collected, embedded into an inferred social network, and used to generate improved movie recommendations (Golbeck, 2006).

The study of information propagation through social networks is another related area of research. The spread and adoption of *social innovations* within real-world communities (Valente, 2005) is of particular relevance as ensuing models can be applied to online environments. For example, a model for the spread of discussion topics in *web logs*, or blogs, is presented in (Gruhl et al., 2004) and the identification of a minimal set of people whose adoption of a new product would maximize the spread of that product through the given social network is described in (Kempe et al., 2003). However, this is mostly theoretical work that has not been applied in working systems. To the my best of knowledge, I am not aware of a recommender system that works directly with the information diffusion models that I propose in Section 3.3.

3.2 Collaborative Filtering

I begin with a brief overview of the collaborative filtering (CF) algorithm as it 1) later provides the baseline comparison for the effectiveness of my proposed approach and 2) helps highlight the shortcomings inherent in many recommender systems. CF operates on the *user-item matrix*, R, where entry $r_{c,s}$ indicates the rating score user $c \in \{c_1, c_2, \ldots, c_m\}$ has given item $s \in \{s_1, s_2, \ldots, s_n\}$. Each row represents all ratings a particular user has made, and each column represents all ratings a particular item has collected. Often, rating scores follow a numerical scale (e.g. 1 to 5 stars) and are explicit, but they also may be inferred from item purchases and other implicit user actions (Schafer et al., 1999). The ultimate goal is to predict the score of empty cells for the *active user*, the user currently requesting recommendations.

CF algorithms are divided into two categories: *memory-based* and *model-based*. I focus on a memory-based algorithm because it is the most straightforward and is widely used. For a complete review of CF, I refer to (Adomavicius & Tuzhilin, 2005).

3.2.1 Memory-Based Algorithm

Memory-based CF algorithms rely on exploiting gaps within the user-item matrix. The intuition is that users who have similar preferences will generally rate items in a similar manner. Therefore, if the active user c has not rated item s, but the recommender system can find similar or correlated users (i.e. *neighbours*) who have, then a rating score can be predicted using (3.1).

$$r_{c,s} = \bar{r}_c + k \sum_{\acute{c} \in \hat{C}} \mathrm{sim(c, \acute{c}) \times (r_{\acute{c},s} - \bar{r}_{\acute{c}})} \tag{3.1}$$

$$\mathrm{sim(c, \acute{c})} = \frac{\sum_{\mathrm{s \in S_{c\acute{c}}}} \mathrm{(r_{c,s} - \bar{r}_c)(r_{\acute{c},s} - \bar{r}_{\acute{c}})}}{\sqrt{\sum_{\mathrm{s \in S_{c\acute{c}}}} \mathrm{(r_{c,s} - \bar{r}_c)^2} \sum_{\mathrm{s \in S_{c\acute{c}}}} \mathrm{(r_{\acute{c},s} - \bar{r}_{\acute{c}})^2}}} \tag{3.2}$$

$\hat{C}$ is the set of neighbours for the active user and implies there are some number of items in common that have been rated by both the active user and each neighbour. Users tend to use ratings scales differently. For example, on a 1 to 5 rating scale, the active user may seldom rate 1 or 5 while a neighbour only rates 1 and 5. Therefore, the average rating of the active user and current neighbour ($\bar{r}_c$ and $\bar{r}_{\acute{c}}$, respectively) are used to smooth out this inconsistency.

The Pearson coefficient (3.2) correlates the degree of similarity $\mathrm{sim(c, \acute{c})}$ between two users where $S_{c\acute{c}}$ is the set of common items both users have rated. The degree of similarity ranges from -1 (perfect negative correlation) to +1 (perfect positive correlation). The similarity value is then used by equation (3.1) as the impact weight each neighbour has in determining the final predicted value (typically the N most similar neighbours are used). Thus, a neighbour with a similarity value 1 will have a large influence in moving the predicted score towards her (relative) rating. Finally, k is a normalizing factor and is the inverse summation of the absolute similarity values.

3.2.2 Limitations and Observations

Conceptually, CF is intuitive and has the advantage that nothing needs to be known about the items in order to make predictions. For items that are difficult to automatically analyze, like video, this is clearly beneficial. However, CF does have a number of shortcomings: the most serious is its sensitivity to the inherent *sparsity* of the rating matrix. Unless users experience immediate benefits to rating items, they are not normally inclined to do so and this leaves large holes in the rating matrix. Besides, in applications where *millions* of items are present, it is impossible to expect that any user will rate a significant percentage. Another serious limitation is when new columns (i.e. items) and rows (ie. users) are inserted into the matrix. There is usually a significant time lapse before enough ratings are built up to either 1) recommend the new item to users or 2) recommend items to the new user. Finally, CF does not scale well when considering very large numbers of items and users as it becomes computationally prohibitive to calculate correlations (Sarwar et al., 2000).

The overall challenge is illustrated by Figure 3.1 and can be seen as identifying a subset of users and items (3) from the entire rating matrix (1) that are relevant to the current recommendation decision, i.e. how would user M rate item N? Ideally and intuitively, this subset only includes users who are like-minded to M and items that are related to N. This scenario has the best probability of yielding the best prediction possible but correspondingly increases the probability user M already knows about item N. For instance, fans of Steven Spielberg are probably aware of all his movies and enjoy the majority of them. That aside, it becomes clear that collaborative filtering is not sufficient by itself. There exists two graph-based techniques that complement it: *spreading activation* (Huang et al., 2004) and *horting* (Aggarwal et al., 1999). Both reduce the sparse rating matrix to a more dense matrix by crawling *transitive* relationships between users' ratings and eliminating users and items that fall outside the crawl. While these techniques do boost prediction accuracy, they do not help with problem of new items and users. It is more common to determine something about the items (e.g. movies directed by Spielberg) and/or something about the users' overall tastes and preferences (e.g. fans of Spielberg movies). *Content-based analysis* is needed if users and items are to be clustered together using some other set of criteria besides ratings. Indeed, many recommender systems combine collaborative filtering with some form of content-based analysis as each addresses shortcomings in the other (like the new user/item problem, etc). Such recommender systems are called *hybird recommender systems* (Burke, 2002). Unfortunately, this increases their complexity and introduces new issues and challenges.

No matter how additional information about items/users is computed and introduced, the goal remains the same: identifying a reasonable subset of related users

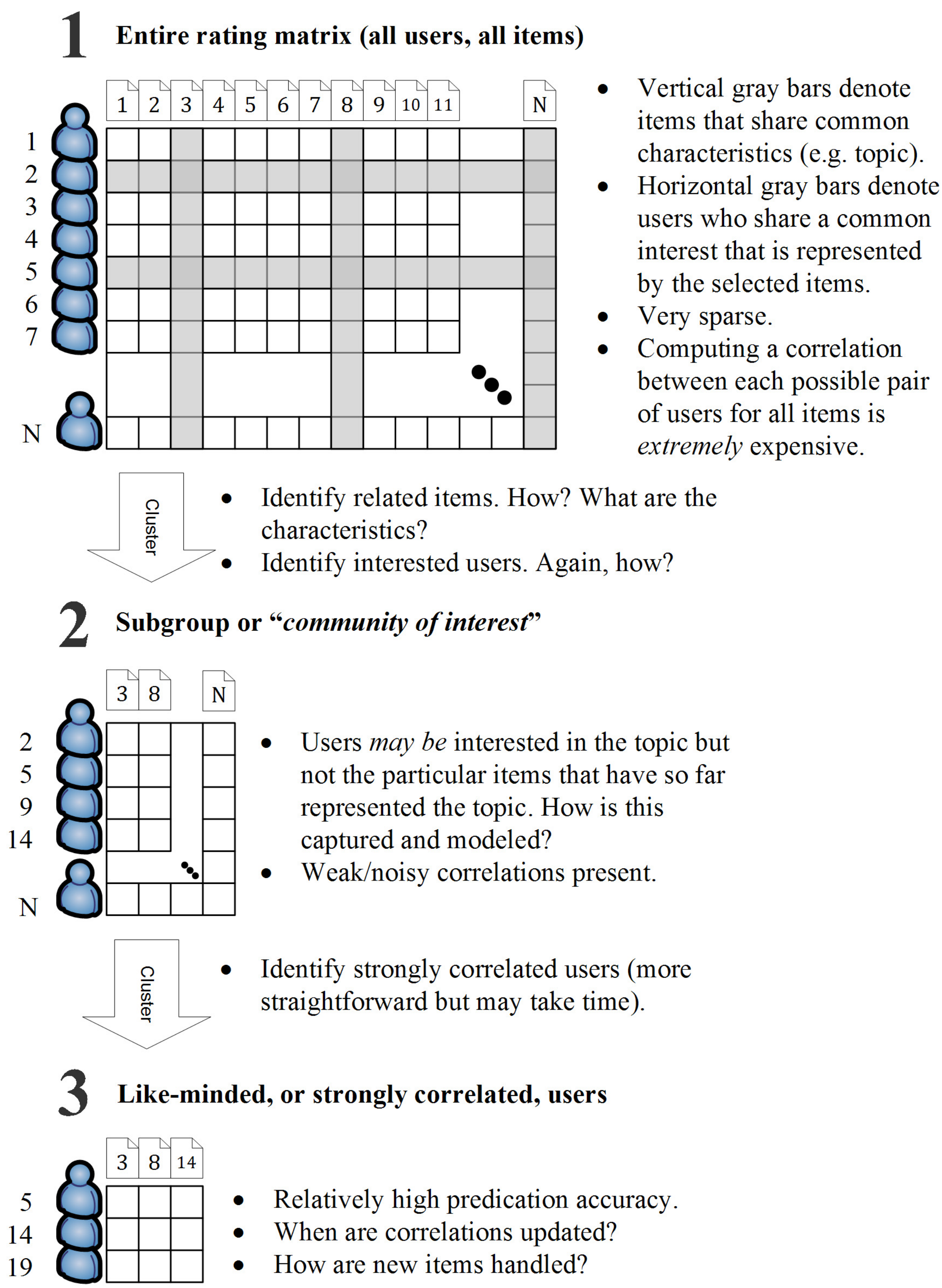


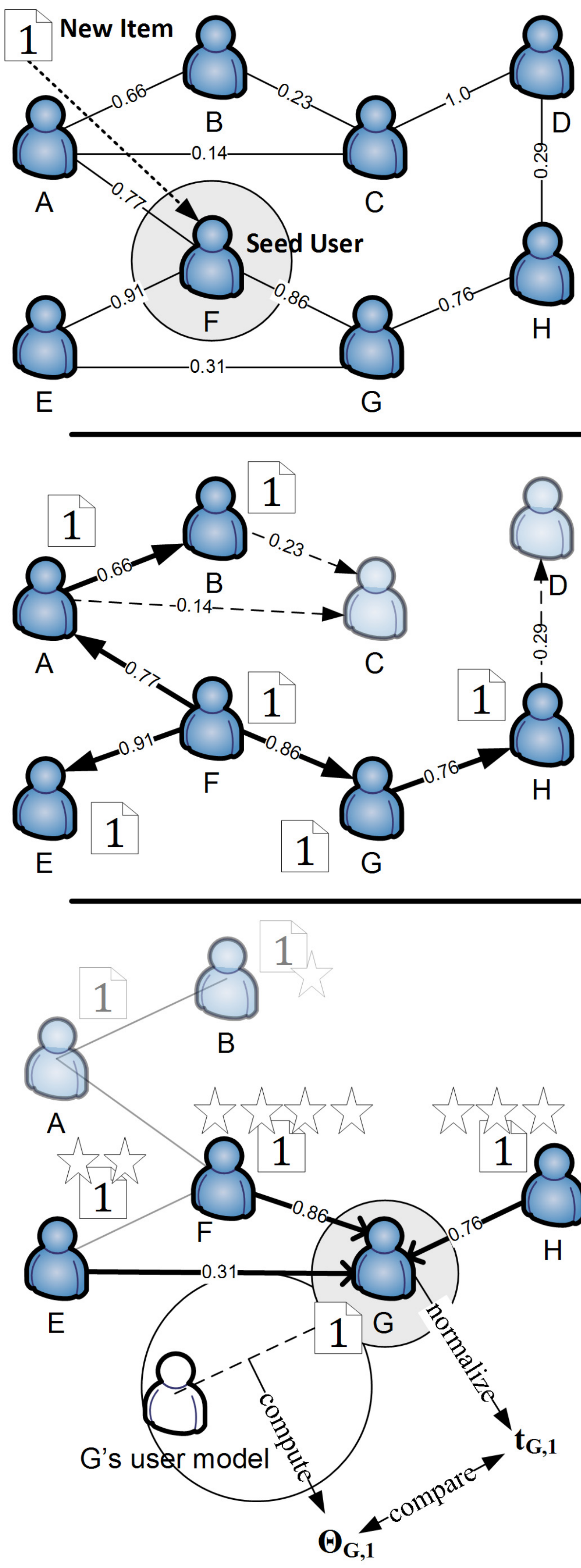
Figure 3.1: Addressing the limitations of collaborative filtering through clustering of users and items.

and items that is depicted as (3) in Figure 3.1. I argue that an intermediate step (2) is missing from the previous consideration. When considering an application where many topics or item categorizations exist, users will not always have their tastes correlate across all topics, i.e. computing correlations from the entire rating matrix (1). And, as it has been shown, it is difficult to narrow the matrix to a specific subset (3) through automatic computation. I believe that (2) represents a comprise between the two extremes: namely, the social context of delivering recommendations is taken into account. Communities of interest are typically formed around specific topics of interest but not all users must necessarily share the same opinion, and an individual user can expect to be subjected to many new and different perspectives, an aspect which makes online communities exciting. In the pursuit of automatically determining (3), the self-organizing system (2) is overlooked and neglected which may be potentially exploited to produce (3). In the next section, I describe the algorithms I use to allow (2) to develop and later exploit to distribute and recommend items to users.

3.3 Social Network Approach: Push-Poll

The term "collaborative filtering" is a misnomer as users never explicitly coordinate with each other to produce (better) recommendations for themselves or others. I propose a "social" recommender system that allows users to coordinate and develop communities of interest and exploits the corresponding social network as the primary method to distribute and recommend information items: the word of mouth process is supported rather than replaced. Also, I place emphasis on dealing with new items and users. This section explains the "push-poll" approach: a term I use to describe the collection of algorithms required to build a more socially-orientated recommender system. Figure 3.2 and the following discussion summarizes the main processes involved in push-poll.

Push-poll models the implicit community of interest that normally develops around a shared topic of interest as a *subgroup* of a larger social network. Fig-



1. SEED
Consider 8 users who are all interested in some topic X, and the depicted network *subgroup* shows the *influence* value between each connected pair of users.

Consider a new item, **1**, that is related to topic X.

Let user **F** act as the *seed* for item 1 and is said to be *infected* with it.

2. PUSH
From the seed user, the item spreads, or is *pushed*, through the subgroup using the *Independent Cascade* model. Ultimately, users A, B, E, F, G, and H are *infected* which means the item is placed in their respective *queue*.

The influence value is used as the probability that one user infects another. Users C & D were not infected although several attempts were made (note the unlikely probabilities).

3. POLL
When G requires a recommendation, her *neighbours* (E, F, & H) are *polled* according to the *Threshold Model of Collective Behavior* for all her items that are currently queued. Note that users A & B are not considered because they have no direct connection to G (review the network in block 1).

The influence value is now used as a weight on the *feedback value* (i.e. rating) from each particular neighbour. All feedback for item 1 is normalized to a single value, $t_{G,1}$ [0,1].

An *activation threshold*, $\Theta_{G,1}$ [0,1], is automatically computed and represents G's *internal resistance* to the item (0-low; 1-high). If $t_{G,1}$ is greater than or equal to $\Theta_{G,1}$ then item 1 is recommended to G.

Figure 3.2: Overview of the push-poll approach.

ure 3.2 shows a subgroup of 8 users interested in topic X. The edges between each pair of users indicates an existing *influence* relationship between them. The edge weight shows the strength of *influence* between a particular pair. A low influence value (approaching 0) indicates if one user likes a particular item regarding topic X, it is unlikely that the other will as well. A high influence value (approaching 1) indicates the opposite. Depending on the user's interests, she may be present in multiple subgroups and have multiple influence relationships with any one other user.

Now consider the case that a new item, 1, has entered the system. It requires some initial content analysis to be matched to appropriate subgroup(s). Since push-poll uses content-based analysis, it can be considered a hybrid recommender system. After the new item's content has been analyzed and deemed to be related to topic X, it is *seeded* into the subgroup as depicted by Figure 3.2. The seed user, F, is modeled to be the person in the social network who initially had the idea/innovation/disease and so forth that the item represents. From the seed, the item spreads or is *pushed* through the subgroup according to an information diffusion model: e.g., if user A has repeated contact with F who has a "contagious" item, then there is a high probability A will also be infected. Thus, the item is more likely to spread between users with high influence values compared to users with low influence values.

The push process is instantaneous: at the end, some or all users in the subgroup will be infected with the item and some will not. If a user, G, is infected with an item, it is placed in her *subgroup queue* and is not yet recommended. When G requests a recommendation, or the system deems it is an appropriate time to make a recommendation, then the *poll* process is triggered. Poll is modeled on a separate information diffusion model from push. For each queued item, poll calculates an *activation threshold*, θ, which represents the G's internal resistance to the item (or natural immunity, etc.). If θ is low or negligible (close to 0), then the system is confident that G will like the item. Conversely if it is high (close to 1), then the system is confident that G will **not** like the item. Poll then looks to the user's neighbours (i.e. adjacent users in the subgroup) for their feedback on the item. If the neighbour has given feedback, then her (implicit or explicit) rating is weighted by the influence value between the users and normalized with all other neighbour feedback to an aggregated influence value, t. If t is greater than or equal to θ, then item 1 is recommended to user as the influence from her neighbours has overcome her internal resistance to the item.

Finally, if G gives feedback on an item, then the influence values between herself and her neighbours who have also given feedback are updated. If the user and her neighbour are in agreement, the influence value increases; otherwise, it decreases. Also, a user in the subgroup who has given feedback but is not G's neighbour may have a new influence relationship created between him and her. Thus, the subgroup is constantly reshaping itself as new content is introduced.

In real life, our social network–the type of relationships we have with others and the strength of these relationships–plays a key role in disseminating personally relevant information. The same is true, to some limited extent, in collaborative filtering except that the different relationships between users are hidden and are not fully taken into account. By basing a recommender system entirely on social networks, it is anticipated that a number of benefits can be achieved. The main benefit being that the self-organization of users can be exploited to generate better recommendations and a more intuitive recommendation process.

The following subsections explore the individual components of push-poll in greater detail. First, I discuss how subgroups are to be represented and the criteria for including users in a subgroup (Section 3.3.1). Next, I examine the information diffusion process (i.e. "push") and how it initially spreads items to users' subgroup queues (Section 3.3.2). Section 3.3.3 shows how the feedback/ratings of other users are to be incorporated into the process of activating (i.e. recommending) items. The final section, 3.3.4, discusses how feedback changes the subgroup, affecting the spread and activation of subsequent items.

3.3.1 Subgroups

Subgroups represent a relatively small network of users who are interested in a particular topic that can be either broad (e.g. "science") or increasingly more specific (e.g. "biochemistry"). Subgroups are enmeshed within a larger social network, hence the terminology, and are meant to capture communities of interest. Users are represented as nodes and an edge between nodes describes that one user *influences* another with a specific strength and vice versa. The edge weight between a pair of nodes is therefore known as the *influence value* (ranging from -1 to +1) and is related to the similarity between the users' preferences regarding the subgroup's topic. Depending on a user's interest, she may be present in multiple subgroups and hold multiple influence relationships with another user.

How subgroups are initially formed is dependent on the recommender system's application. For instance, if the system is an online discussion forum, such as Comtella Discussions, a specific forum may define a subgroup and posting to the forum may deem that the contributing user is "interested" in the topic of the forum and should be included in the subgroup. Or, a subgroup may be quite fluid, e.g. all users who have a certain keyword repeated N times in their posts. I propose the simplest approach is to simply let users explicitly create, join or leave a subgroup as they wish (in keeping with being more social), and the initial influence value between users can be the Pearson correlation of any subgroup-related items that have been rated. If not enough rating data exists, then the influence relationships and values can even be generated randomly. Again, the goal is that subgroups are ultimately self-organizing, so a user who is randomly "woven" into a subgroup should have her relationships and their respective influence values quickly adapt to her real preferences.

Allowing users to explicitly create subgroups is in keeping with the philosophy of collaborative tagging systems: let users organize content themselves and do not impose a pre-determined hierarchy, etc. Chapter 4 details my implementation of the push-poll that allows subgroups to be defined in multiple ways and one method is by listing a set of tags, e.g. *politics foreignpolicy*.

3.3.2 Push (Diffusion)

Push is the process which distributes items to users within a certain subgroup. Consider that a new item has entered the system and content-based analysis is performed on the item to determine its content. If the item is a text document, I suggest that extracting its significant terms is sufficient enough to enable a rough guess as to what subgroup(s) the item initially "fits" into (if subgroups are defined by tags as previously mentioned). In other cases, different techniques would be needed for other content items such as video or sound (e.g. music files would require an algorithm capable of anlayzing the number of beats per minute, etc.). After a new item has been matched to one or more appropriate subgroups, it is *seeded* into each subgroup: a small number users are chosen to be the initial *seeds* of the item, i.e. the nodes that are considered to have initially originated the item. If the item is being contributed to the system by one of its users, then that user can obviously be the seed node; otherwise, I suggest some criteria for determining a potential "surrogate" seed: the user provides quick feedback (e.g. rates often) and acts as an authority (i.e., exerts strong, direct influence on many users). Seeds could also be chosen completely at random; however, I leave seed determination as future work.

After seeding, the influence value between user pairs determines how items will propagate through the subgroup as explained by the *Independent Cascade* model (Goldenberg et al., 2001) that captures the probability a person will choose to *adopt* an item depending on how many of her social contacts have already adopted it (note, the item could be a new hairstyle, gadget, etc.). I use the Independent Cascade model to spread items across the subgroup but modify the terminology to illustrate that users have no voluntary control over whether they adopt an item in the push process or not. Instead of "adopting" an item, a user is *infected* with it, and infection is a condition where the item becomes a candidate for activation (Section 3.3.3). At the start of a push, all seed nodes try to infect their "contacts", or neighbour nodes (i.e. the nodes at the end of outgoing edges), with the item. A seed node u infects a neighbour node v with probability $p_{u,v}$–the absolute value of the influence value from node u to v. Infected nodes have a single attempt that will either succeed or fail at infecting a neighbour node. Success or failure is independent of all previous attempts to infect the node in question. Note, this assumption is relaxed in the *General Cascade* model (Kempe et al., 2003). After the seed nodes cannot induce any new infections, all newly infected nodes try to infect their neighbours, and this breadth-first cycle repeats until no new infections are possible. Ultimately, depending on their direct/indirect connections to seed users, some users in the subgroup will be infected while others will not.

3.3.3 Poll (Activation)

If a user is infected with an item, the item is left in the user's respective *subgroup queue*. Poll is the process that ultimately activates (i.e. recommends) these queued items, and it is based on the *Threshold Model of Collective Behaviour* (Granovetter, 1978). This model describes that node v has an intrinsic threshold level $\theta_{v,s} \in [0, 1]$ for adopting item s and a set of contacts I that have already adopted. For each node u in I, there is an associated weight $t_{u,v}$ that describes how much "influence" u exerts on v.

Node v will adopt s if (3.3) holds true, i.e., the influence exerted by v's contacts is greater than v's internal resistance to adopting s. In many models, θ is randomly chosen from a distribution (uniform) to capture various levels of willingness. In this case, I is the set of infected neighbours and θ is computed as a confidence level based on some type of content analysis (e.g. comparing how similar the item's significant terms are to previously liked and disliked tags by v). If the system is confident that the item is relevant (e.g. $\theta < 0.25$), then the item is automatically activated. Otherwise, the active user's infected neighbours are polled using (3.4).

$$\sum_{u \in I} t_{u,v} \geq \theta_{v,s} \tag{3.3}$$

$$k \sum_{u \in I} t_{u,v} \times r_{u,s} \geq \theta_{v,s} \tag{3.4}$$

Equation (3.4) is similar to the CF prediction (3.1) except rating scale smoothing has been dropped and influence strengths between nodes are used instead of Pearson correlation values (3.2) which is a computationally expensive operation. The rating value $r_{u,s} \in [-1, 1]$ captures explicit feedback on the extremes (that u did or did not like the item), and implicit feedback lies on medium values following Nichols' implicit rating strength order (Nichols, 1997). Note, the normalizing factor k allows incoming influence strengths to sum to values greater than 1.

Determination of θ and polling is only performed when needed, i.e. when the user is active and is requesting recommendations for the specific subgroups(s). There is a definite timing issue to this approach as users activating an item early in its lifetime will find infected neighbours have not yet provided feedback. One workaround would be to automatically activate the item for seed nodes, assuming these users will most likely see the item first. Otherwise, an item that failed to be activated could be saved back in the queue for a later activation attempt.

3.3.4 Network Feedback

Once feedback from a user for an item is recorded, influence values with neighbours who have also provided feedback are updated. Feedback can be implicit (e.g. following the link of an item to the full story) or explicit (e.g. tagging an item). Note, if feedback is explicitly positive, then a "re-push" could be triggered using the active user as the new seed node. Users in agreement will see their influence values move to either positive or negative unity while users with low/noisy agreement will have their connections dropped. Network readjustment will ultimately affect the subsequent spread and activation of later items. In this instance, a simple pay-off scheme could be used to adjust influence values and smooth out any wild variations in agreement. However, more advanced learning algorithms could be used instead.

3.4 Evaluation

I compare the performance of a basic implementation of push-poll to the CF algorithm reviewed in Section 3.2 using a simulation. My goals are to show that the social network approach of push-poll is feasible and to gain insight into the advantages/disadvantage of the approach.

I used the well-known *100K MovieLens* data set which contains 100,000 ratings (on a scale of 1 to 5) by 943 users for 1682 movies (GroupLens, 2003). Each user is guaranteed to have rated a minimum of 20 movies. Data was captured during a 7 month period from September 1997 to April 1998.

The metric, mean absolute error (MAE), is used to compare performance.

$$\mathrm{MAE} = \frac{\sum_{\mathrm{u=1}}^{\mathrm{N}} |\mathrm{r_{u,i}} - \acute{\mathrm{r}}_{\mathrm{u,i}}|}{\mathrm{N}} \tag{3.5}$$

N is the total number of rating-prediction pairs attempted, $r_{u,i}$ is the actual rating given by user u on item i, and $\acute{r}_{u,i}$ is the predicted rating. Over- and under-estimation of $r_{u,i}$ by $\acute{r}_{u,i}$ is treated the same by taking the absolute value of the difference between the two. A lower score means more accurate predictions.

Descriptions of the movies' plot and acting/production crew were not included with the data set. Therefore, only a small amount of content analysis was used, i.e. movies were categorized by their genre. Activation thresholds were not calculated as the actual rating given by the user is being predicted.

My hypothesis is that push-poll will perform as well as or better than CF at predicting ratings. In a general system, it is anticipated that the number of users in any give subgroup will be small. Therefore, I wish to investigate how push-poll performs in small vs. large user groups. I also hypothesize that push-poll will do better in specific topics (e.g. *biochemistry* vs. *science*) with a small group of highly interested users as stronger influence relationships are more likely to develop in such situations.

3.4.1 Simulation

I chose to classify movies by genre due to the lack of additional information in the data set, such as plot summaries. A general and a specific genre classification were selected to represent subgroups a user could "join": {*adventure*} with 135 matching movies and {*science-fiction action adventure*} with 27 matching movies, respectively. For example, *The Princess Bride* (*action, adventure, children's, romance*) would be included in the general genre subgroup but not the specific genre subgroup. I also wanted to test 2 different subgroup sizes: one with a large number of users versus one that has relatively fewer users. Thus, a minimum number of genre subgroup movies must have been rated before a user is considered to belong to the respective subgroup. When this number is set low, a large number of users (~ 200) are considered to belong to the subgroup. When the number is set relatively higher, the subgroup shrinks to a smaller number of users (~ 25). Finally, there is the question of how much training a recommender system requires before making accurate predictions. The *training set* is comprised of items whose ratings are already known by the algorithm(s) being tested, while the *test set* is comprised of items whose rating will be predicted by the algorithm(s). I have elected to use 2 different training/trial set sizes: one that has a relatively large training set (80% of the subgroup movies) and one that has a relatively small training set (20% of the subgroup movies). Altogether, there were 8 simulation configurations (2 genre subgroups * 2 subgroups sizes * 2 training/test set sizes) with 5 random test sets run 5 times apiece for each configuration (i.e. 5-fold cross validation). The MAE for each of the 25 runs were averaged, as reported in the next section. Table 3.1 lists the simulations.

For push-poll, seed nodes were randomly selected from subgroup users for each test movie. System parameters for push-poll were optimally set depending on the number of users in the subgroup: the number of seed nodes was set to ensure the majority of users were infected (~ 10 users for both large and small subgroups), and the number of infected neighbours polled was 10% of users for the large subgroup and

Table 3.1: Simulation sets.

Simulation	1	2	3	4	5	6	7	8
# of users	200	200	200	200	25	25	25	25
Genre: General(G)/Specific(S)	G	G	S	S	G	G	S	S
Total # of movies	135	135	27	27	135	135	27	27
# of training movies	108	27	22	5	108	27	22	5
# of test movies	27	108	5	22	27	108	5	22

20% for the small subgroup. Push-poll requires initial influence values between users. The Pearson correlation values (3.2) calculated from the ratings matrix with non-subgroup users, non-subgroup movies, and test movies removed were used as initial influence values. However, CF was allowed to use rating information from movies outside the current subgroup in addition to what push-poll used–significantly more information (i.e. the ratings given to roughly 800 more movies).

Because an actual rating is being predicted, (3.1) was used by push-poll with influence values substituted for Pearson correlation values (activation thresholds were not determined). Each test movie had all its predictions for users who had rated it performed sequentially (by the rating timestamp) before the next test movie was seeded. The influence value between a pair of users was updated if it was determined that each user had rated the same test movie. CF performed predictions in the order of the rating timestamp, regardless of the test movie. Finally, after a lapse of 24 hours[4], CF was allowed to update the Pearson correlation values between users using any new rating information introduced between lapses (these re-calculations took the bulk of the simulation time as many days may elapse before a specific movie received a new rating).

[4]Recommender systems that use collaborative filtering often compute Pearson correlations at night when system demand is at its lowest.

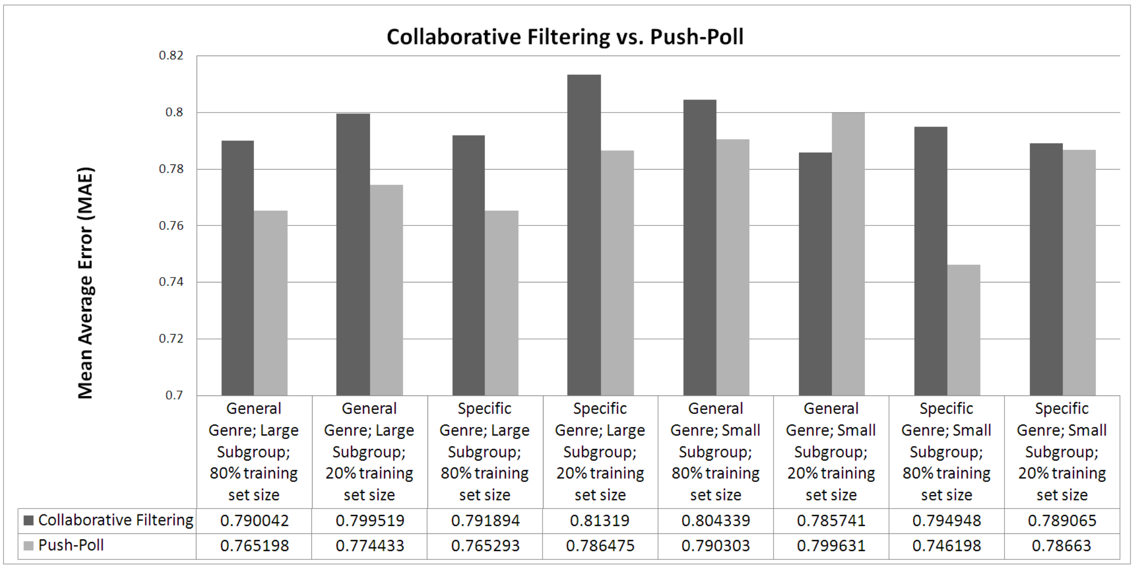


Figure 3.3: Mean average error (MAE) results for simulation sets.

3.4.2 Results

Overall, push-poll significantly outperformed the CF algorithm's MAE score by an average of 1.93% ($p<0.001$). This is an encouraging result as it shows that the extraneous rating information that was used by CF should not have been considered and that subgroups maintain "good" influence values between their users.

The results for the "large" subgroup are presented in the first half of Figure 3.3. On average, push-poll consistently and significantly outperformed CF by 2.58% in simulations where there are relatively more users present in the subgroup and both algorithms performed better with the larger training set (the 80%/20% configuration). However, this intuitive expectation is reversed for CF in subgroups with fewer users: I believe correlations for a small number of users are noisy when considering all rating information and prediction accuracy is largely dependent on the selected test movies (i.e. popular movies that have lots of ratings versus relatively obscure movies that have few). Yet, push-poll's behaviour remains consistent for the training/test splits but experiences increased variance in its scoring. I hypothesize that at the time of rating a user may find only a few neighbours who have already rated and their influence values are low. In such a scenario, a complete implementation would leave that movie in the queue, waiting for feedback from stronger connections.

According to the hypothesis, for small subgroups, push-poll performed better when considering the specific genre compared to the general genre (an average of 1.1%). Its best performance (.7462) was in the simulation where a small subgroup of users were rating movies with the specific genre (with a large training set), indicating that careful selection and development of influence relationships between like-minded users within a subgroup will lead to improved prediction accuracy.

3.5 Summary

In this chapter, I presented the design of a push-poll recommender system that supports word of mouth processes by distributing and recommending information items through subgroups of social networks. A basic implementation of my algorithm significantly outperformed a common CF algorithm. There are a number of advantages to this approach: 1) recommendation is modeled on real life processes and not as an outcome of pre-arranged rules, giving users some intuition over how their interactions affect which items are recommended to them, 2) new items are introduced with a minimum of content analysis (although there is some); and, 3) the underlying algorithm is computationally efficient since a Pearson coefficient is not computed (influence values are looked up on demand and updates on the value due to feedback are a O(1) operation). The next chapter describes the implementation of the push-poll approach in recommending news articles.

Chapter 4

Towards Supporting Online Communities

In this chapter, I expand on the work presented in Chapter 3 and describe a working recommender system, KeepUP, that uses the push-poll approach. Since items are "pushed" through a social network subgroup, the presence and strength of edges between users are crucial factors in determining recommendations. Thus, one part of this chapter will focus on developing an interactive visualization that allows the active user to view her neighbours (i.e. adjacent users). In addition to displaying the degree of influence each neighbour exerts on the active user's recommendations (and vice-versa), the active user can manually adjust neighbours' influence, triggering KeepUP to instantly "re-recommend" a small set of items which appear along with the visualization. This is similar to work done in Aimeur & Mani-Onana (2006) where e-commerce users are allowed to restrict the collaborative filtering process to a set of manually selected contacts each of whom has a level of credit or trust that is factored into the final recommendation of items. It was shown that these "local" recommendations were better than those made by unrestricted k-nearest neighbour collaborative filtering[1].

In the previous chapter, I demonstrated that the push-poll approach outperforms a straightforward collaborative filtering algorithm in a simulation of predicting user ratings on movies. However, predicting item ratings on a numerical scale is not the objective of push-poll. It is meant to direct new information to interested users who can then collaborate on further classifying the information. Also, a valid critique of such an evaluation is whether users actually notice a difference within an

[1]This chapter is based on an earlier work: The KeepUP Recommender System, in Recommender Systems 2007, Andrew Webster and Julita Vassileva, {to appear (October 2007)} ©ACM, 2007.

(albeit small) improvement. I propose a multi-stage evaluation that begins with investigating the effectiveness of push-poll at building and maintaining implicit social network subgroups, distributing RSS items through these subgroups, and making recommendations. Next, I intend to evaluate what impact the visualization has on user behaviour, whether users find it beneficial, and if prediction accuracy tends to increase due to manual influence adjustments.

4.1 Overview of KeepUP

KeepUP[2] is a RSS (Rich Site Summary) recommender system. RSS is a popular method to publish content to the web and is often used by blogs and news services to alert subscribers to new entries. RSS items follow well-known XML formats and usually include a headline, a short description, and a URL to the full item of interest. A RSS feed is simply a web accessible XML document that contains 1 or more items and is updated regularly. The breadth of topics and overwhelming number of RSS feeds presents an exciting challenge for a recommender system that must manage many new and diverse items per day. After a 90-day period, over 220,000 items have been indexed by KeepUP, yet there are only approximately 110 registered RSS feeds. The name “KeepUP” implies that users are able to “keep up-to-date” with personally relevant news and events.

4.2 Channels

Channels are the building blocks of KeepUP and are a user-friendly name for the term “subgroup” which has been used until now. A channel defines the “limits” of a topic that users are interested in, and there are three types of channels that users can create to express their particular interests:

1. Feed Channel: collects items from select RSS feeds.

[2]http://keepup.usask.ca

2. Tag Channel: collects items that match a set of specified keywords, or tags. For example, the channel in Figure 4.1 is based on the tags *climate change*, *Kyoto*.

3. Person Channel: collects items that were rated positively by specified users.

Each type of channel suits a different information-gathering purpose. For example, imagine a user who has a number of favourite web sites and would like to stay current with their updates. This user may want to have the respective RSS feed of each web site grouped into a single *feed* channel that will show the new items appearing at any of the web sites. Or, a user may be more interested in a specific topic and less concerned where the information comes from. In this case, a *tag* channel would be more appropriate. The tag channel will select all items that contain the matching tag or tags regardless of the RSS feed they come from. Finally, a *person* channel allows users to see what items are liked by their friends and colleagues.

Channels display recommended items as a list of headlines (Figure 4.1), and multiple channels can be displayed on a single page. Note that all items appearing in a channel have been recommended (Section 2.3) to the user. Users can then quickly scan each channel for items of the most interest.

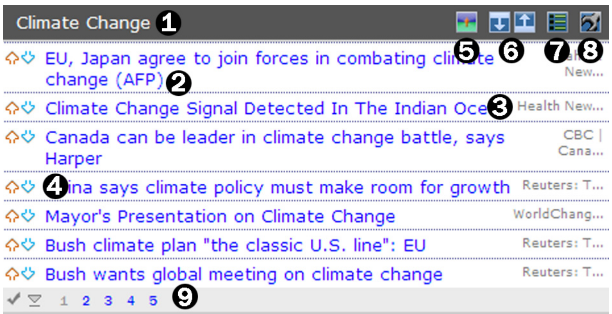


Figure 4.1: An example channel regarding *climate change*.

The highlighted areas in Figure 4.1 are described as follows:

1. Channel Title: user-defined title.

2. Item Headline: in its collapsed state, an item shows only its headline. Clicking a headline expands the respective item (Figure 4.2) and marks the item as

Figure 4.2: An expanded item in the channel shown in Figure 4.1.

read.

3. Item Source: truncated name of the source RSS feed.

4. Rate Item: users can immediately rate an item positively (up arrow) or negatively (down arrow).

5. Channel Neighbours: indicates that there are other users sharing the current channel. Clicking the icon takes the user to the interactive neighbour visualization (Section 4.3.1).

6. Expand or Collapse All Items: a shortcut to expand or collapse all the items (does not mark the items as read).

7. Sort Items: items can be sorted by recommendation, date, title, or web site (i.e. RSS feed).

8. Channel Options: expands to reveal additional options including deleting the channel, setting the maximum number of items to display at a time, etc.

9. Misc. Options: the user can mark all items as read (checkmark icon), refresh the channel (triangle icon), or move to a different page to see more items.

The highlighted areas in Figure 4.2 are described as follows:

1. Link to Full Story: opens a new web browser window which displays the complete story.

2. Link to Item Source: opens a new web browser window which displays the feed's associated web page.

3. Time of Index: the amount of time that has elapsed since KeepUP first indexed the item.

4. Item Description: depending on the feed, varies from a single sentence summary to the complete story including graphics, videos, etc.

5. Add to Favourites: bookmark items for later reading.

6. Item Tags: the 5 most popular tags currently applied to the item (users are encouraged to add their own).

Whenever a user creates a new channel, it is available for all other users to "subscribe" to (Figure 4.3). For example, multiple users may join the Climate Change channel of Figure 4.1, adding it to their list of channels. When this occurs, the newly subscribed user is "woven" into the existing subgroup of currently subscribed users (see Section 3.3.1). This process involves computing an initial Pearson correlation value between the new user and each existing user to use as the edge weight (i.e. influence value) between each pair. If there are not enough previously rated items in common to perform a Pearson correlation, then I compare user profiles for similarity. Finally, if no edges can be established, then the new user is randomly connected to a subset of the existing users. It is important that the new user have a least one incoming edge from the network to begin receiving items within the respective channel. In a complete system, there will be privacy controls which will allow users to anonymously create or join channels. For now, I ignore privacy issues.

4.2.1 Push-Poll Implementation

There is a dedicated process within KeepUP that continually scans the list of registered RSS feeds, looking for new RSS items. This list of feeds can be set by

	Channel	Type	Started By	Description	# Users	Created
join	**Geek News**	(Feed) 'Engadget' 'Gizmodo' 'Slashdot'	[redacted]		2	May 02 07
join	**Autism Channel**	(Tag) autism	[redacted]	Autism News	1	May 02 07
join	**Microsoft Silverlight**	(Tag) silverlight	[redacted]	Microsoft's answer for the interactive web.	2	May 02 07
join	**SOA II**	(Feed) 'Sandy Carter: S...'	[redacted]		1	May 02 07
join	**ESB**	(Feed) 'ebizQ - Enterpr...'	[redacted]		2	May 02 07
join	**Madmuc's Channel**	(Tag) madmuc	[redacted]	Tag stories with 'madmuc' if relevant to others in the Madmuc lab	2	May 02 07
join	**Social Computing and Web 2.0**	(Feed) 'Many-to-Many' 'Musings about T...' 'Scobleizer: Mic...' 'Weblogg-ed'	[redacted]		4	May 01 07
join	**Tech News**	(Feed) 'Ars Technica' 'CNN.com - Techn...' 'Google News Can...' 'TechCrunch'	[redacted]	New gadgets, software, cool gizmos.	1	Apr 24 07
join	**The Interesting Channel**	(Tag) interesting	[redacted]	Tag a story with 'interesting' if you wish to share that story with others in this channel	7	Apr 13 07

Figure 4.3: A list of channels that users can potentially join–each channel is displayed with its type, who started it, an optional description and the current number of subscribed users.

the developer or freely added to by users, depending on the application. For instance, feeds could be restricted to certain URLs, such as popular blogging websites (thus, only blogs entries are allowed). What feeds are registered sets the "tone" and "tempo" of the community, and with KeepUP only a dozen popular RSS feeds were initially registered, the rest can be set by users. When a new RSS item is detected, I suggested earlier in chapter 3 that text documents require only a trivial amount of content-based analysis in order to be classified: term extraction of the RSS item's headline and description enables a rough guess as to what tag channel(s) the item initially "fits" into. These initially extracted terms are called the *system tag set* for the item. Later, tagging by users triggers a re-examination of the item, possibly causing it to be introduced into other channels. Of course, the item's feed immediately dictates what feed channels the item is to be pushed into.

According to Section 3.3.2, the item is pushed into the selected tag and feed channels. Users, who subscribe to one or more of these channels and are "infected" with the item, will have the item placed in their respective channel queue(s). Once a user requests that a channel be updated (i.e. recommendations are to be made), all items in that user's channel queue are subjected to the poll process of Section 3.3.3. First, an activation threshold is calculated for each item by comparing the item to the user's profile. In KeepUP, user profiles are very simple. Whenever feedback is detected (e.g. user clicks on a item's headline, rates an item, etc.), the set of tags that are currently applied to that item[3] are added to the user's profile as a keyword vector. The vector is assigned a preference indicator (i.e. did the user like or dislike the item depending on the feedback?). To assign an activation threshold, KeepUP looks for vectors in the user profile that are similar[4] to the current item's tag set. If a reasonable match is found, then the activation threshold represents KeepUP's confidence that the user will either like or dislike the current item based on feedback of previous items that are similar. If the item is very similar to items that the user has previously liked or disliked, then the item can is immediately recommended or discarded, respectively. Otherwise, the poll process looks to the user's neighbours for their feedback on the item. Their feedback is weighted by their respective influence value over the current user and then aggregated and normalized to a single value (see Section 3.3.3). If this value is greater than the activation threshold, then the item is recommended (i.e. displayed in the channel), otherwise it is discarded from the queue. In the scenario where there is not enough information to calculate either an activation threshold or an aggregated neighbour influence value, then the item is placed back in the queue for later activation. If an item stays in the queue for too long (e.g. two weeks), then it is discarded.

User feedback does not have to be explicit. In KeepUP's case, implicit feedback is recorded whenever a user interacts with an item, e.g. when the user clicks an item's headline. This provides much needed information when polling new items to recommend as it increases the probability that there are some neighbours who have previously given feedback on the item. There are also some special considerations for certain kinds of feedback, such as when a user marks an item as a "favourite."

[3]These tags may be the initial ones assigned by KeepUP's term extraction, ones that were added by users later, or a mix of the two.

[4]Similarity is calculated by computing the cosine of the angle between the two vectors (i.e. the two tag sets).

This action saves the item to the user's favourite folder but tells KeepUP this item is of considerable worth to the user. Thus, the item is "re-pushed" to *all* of her neighbours and not just the neighbours in the channel that the item appeared in. It is anticipated that this re-push will help spread items of "serendipitous" value throughout the KeepUP community and expose users to new topics.

4.3 Towards Supporting Communities

I believe recommender systems have a large, *supportive* role to play in the exchange of information between users, and one of my arguments from chapter 3 was that recommender systems should and can be more "social." One potential opportunity is to consider a large online community such as MySpace where a single user will only ever see a tiny fraction of all available content. Collaborative filtering is difficult to perform at the level of millions of users and items as was highlighted in Section 3.2. However, push-poll offers a strategy of targeting subgroups of users (i.e. communities of interest) within a larger social network and that good recommendations can be generated for these subgroups. The goal is to make users confident that while they are not actively searching for information items, personally relevant items are continually searching for *them*, especially items from "unknown" parts of the network. KeepUP is partly a response to this challenge and takes initial steps in this direction. I believe that the design and structure of channels lets users self-organize and play a key role in the spread of information. Tagging allows for a shared vocabulary to emerge in the self-organized communities that spring out of users creating and joining channels of their choice (Sen et al., 2006).

When deciding which channels to add, users are presented with a choice. One consideration of this choice is that a single tag, feed or person can exist across multiple channels. For example, in a system with hundreds of users, the possibility of a popular feed, e.g. Slashdot, appearing in more than one channel is high. It is likely that Slashdot is mixed in with a number of other science and technology-related feeds. The user can choose the mix that contains her most preferred feeds (thereby joining a channel where the preferences of the existing users are more aligned with hers). Alternatively, the user can create an entirely new feed channel that uses Slashdot mixed with other feeds, for example, a feed of a little-known blogger who often discusses Slashdot articles (thereby creating a new association between Slashdot and the unknown blog), thus expanding the diversity of channels and the choice-options for other users.

Another consideration that a user makes in choosing a channel is the number of users who are subscribed to the channel. A user may choose to join a channel that has some undesirable tags, feeds, and people, but has more subscribers than other similar channels, assuming she believes a channel with more users results in better recommendations (in most cases this should hold true). Therefore, the user is exposed to some content that she normally would not consider interesting, but which may turn out to be interesting and useful.

Tagging and tag channels are another form of self-organization as users evolve a shared vocabulary (Sen et al., 2006). For instance, the Interesting Channel in Figure 4.3 is based on the tag *interesting*. KeepUP is tasked with initially tagging new RSS items; however, it only considers significant terms present within the item's text as potential tags. It is unlikely the term "interesting" would be considered significant, and items would not appear in the Interesting Channel automatically. Therefore, users must "power" the channel themselves by tagging items that appear in *other* channels as *interesting*. KeepUP then automatically pushes the newly tagged item into the Interesting Channel. And, as users provide feedback on what they personally find interesting and not interesting by rating the items, I hypothesize the channel subgroup will adapt accordingly, forming clusters which represent the desired subset of like-mined users and related items (see part 3 of Figure 3.1). In turn, these subsets of users could be given extra support (e.g. their own discussion space, incentives to contribute additional information, etc) or targeted with specific items for their feedback.

Unfortunately, KeepUP does not offer means for users to communicate with each other (e.g. forum or commenting system). A discussion system based on the prin-

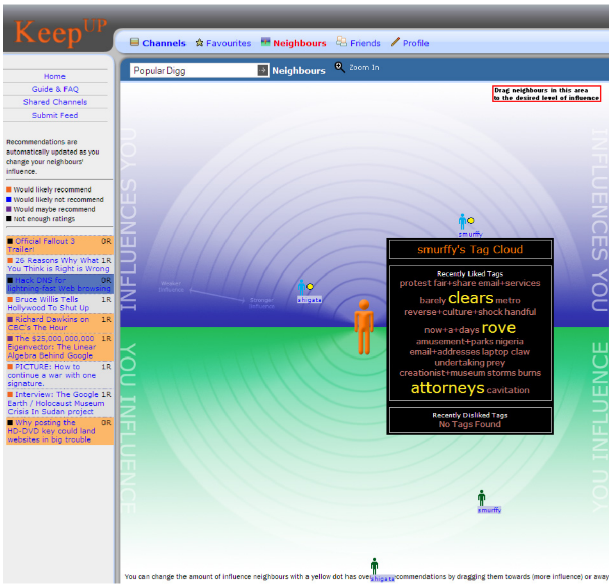


Figure 4.4: Old visualization example with 2 neighbours.

ciples of push-poll would be an interesting avenue for future work. For example, discussion regarding a certain item could be confined to one subgroup and as the discussion builds and evolves, more and more potentially interested users could be made aware of the discussion by the system.

4.3.1 Neighbour Visualization

The presence and strength of edges between users in channel subgroups are crucial factors in determining recommendations. Therefore, I believe it is important for users to be aware of their network "position" and allow them to make manual adjustments to the influence value of incoming edges. Again, I am looking for opportunities to make the recommendation process more scrutable and more social.

Figure 4.4 shows an instance of the interactive neighbour visualization for the channel *Popular Digg* (all items appearing in this channel come from *Digg*[5], a popular URL-sharing community). The visualization is separated into two areas: the top area depicts the amount of influence the active user's neighbours are exerting on her (the closer to the center figure, the more influence); the bottom area depicts the amount

[5]http://www.digg.com

of influence the active user is exerting on her neighbours. Neighbours appearing in the top area (person icon with a little dot) can be dragged within the top area to the desired level of influence (neighbours in the lower section cannot be moved, i.e. a user cannot adjust the amount of influence she exerts on others). As the active user drags a neighbor her set of recommendations along the left-hand side are automatically "re-recommended" based on the new influence value. This allows the user to see what impact individual neighbours are having on her recommendations and whether this impact is desirable or not.

Finally, if the active user holds the cursor over a neighbour's pseudonym, a tag cloud is shown for that neighbour. The tag cloud shows the set of recent tags that are liked and disliked by the neighbour and the relative degree of preference to each tag (i.e. tags in larger font are liked/disliked relative to tags in a smaller font). The tag cloud gives the active user additional information about the neighbour and whether the influence value of that neighbour should be adjusted. For instance, in Figure 4.4, the example tag cloud shows the neighbour, smurffy, has recently liked a number of articles concerning an American political scandal. If the active user is also interested in this particular story, then smurffy could be granted greater influence within this channel, increasing the probability that future items concerning the scandal and other items that smurffy is interested in will be recommended to the active user.

4.4 Ongoing Evaluation

My evaluation study is slated to involve 20-30 self-selected participants who are each asked to rate RSS items that fall under 4 broad topics: arts & entertainment, world, science & technology, and sports news. At the time of writing, I have run 13 participants through the study (primarily graduate students: 6 females and 7 males, ages 19 to 33). The purpose of this evaluation is to collect two types of user feedback: quantitative (i.e. ratings data on articles) and qualitative (i.e. subjective responses to my questionnaire). The former will be used to evaluate KeepUP's objective recommendation accuracy (e.g. recall and precision measures); this aspect of

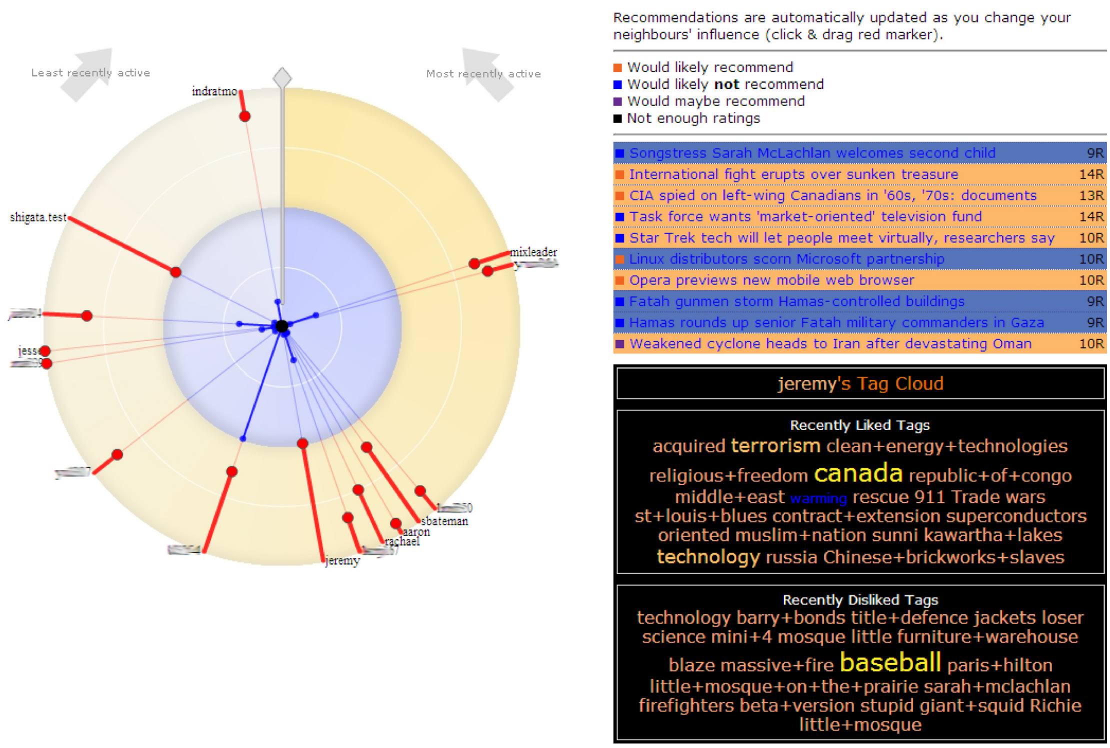


Figure 4.5: New visualization example with 15 neighbours.

the study has yet to undertaken as more participants are required. However, some the qualitative data that has been collected so far is reported in Section 4.4.2. The purpose of the questionnaire is to determine how participants react to 1) KeepUP's recommendation ability and 2) the visualization. Specifically, I am investigating if participants' perception of KeepUP's recommendation ability is positively or negatively impacted after manipulating the visualization (and if the objective numbers match this perception).

Before user testing, I made changes to the neighbour visualization shown in Figure 4.4 with the final result depicted in Figure 4.5. These changes were made to better reflect that the current user is influenced by, and has influence on, others (and what these exact values of influence are). The previous visualization did not seem to express this very well. The current user is depicted as the black dot in the middle of the circle. The inner blue circle shows the influence the current user has on her neighbours. The more influence the current user has on a neighbour, the further the respective blue "ray" grows outwards from the center dot. The outer yellow ring shows the amount of influence neighbours have on the current user. The more influence a neighbour has on the current user, the closer the respective red "ray" pulls inward to the center dot. Users now change influence from others by dragging the red marker "dot" towards or away from the center of the circle. The current user cannot manually adjust her influence on her neighbours (i.e. blue rays cannot be moved like the red rays). Also, there is now a clock metaphor behind the positioning of users (i.e. users on the right hand side have more recently rated articles than those who appear on the left hand side). The tag cloud now appears when a user's name is clicked and stays open while the current user performs other tasks. Thus, the new version of the visualization presents more information to the user (how recent is the activity of other users) and is hopefully more intuitive than the previous version, since the outward and inward rays are two aspects of the same relationship represented by the straight line connecting the pair of users. In contrast, the previous version of the visualization represented each user twice–once in the space of users influencing the current user and once in the space of users being influenced by the current user, which is harder to relate together. I choose to show the evolution of the visualization to highlight how difficult it is to choose the "correct" visual representation that engages users while expressing complex ideas. Finding the correct representation requires a great deal of user feedback and fine-tuning.

4.4.1 Methodology

Unfortunately, there is no formal evaluation framework that can be used to compare recommender systems to each other, and the evaluation of individual recommender systems is a mix of various techniques and chosen metrics (Mirza et al., 2003). I briefly summarize the general structure of the study only to help contextualize the initial results (a deeper discussion of the specifics would be included in a full report). First, participants must rate a small *training* set of articles (20 in all, randomly selected from a pool of 100 articles). They are not required to read the full article just the RSS headline and one sentence summary that KeepUP provides. Participants are also asked to tag at least 5 to 10 of the articles at their discretion. Using this training set, KeepUP then makes recommendations from the 80 articles that remain (the *test* set). Participants are allowed to see all 80 articles separated into two sets (those that are recommended and those that are not) and are asked to rate only those articles they feel strongly about. Whether participants tag articles or not is dependent on their own natural inclination. Next, the participant is shown the visualization from Figure 4.5, given a verbal explanation, and asked to adjust her neighbours' influence in such a way that optimizes KeepUP's "re-recommendation" of the small set of articles that appears along the right side. Each participant is cautioned she may not be able to get a perfect match but is given 5 minutes to do her best. The neighbours in this instance are previous participants (the first three were pilot participants and therefore skipped this part because they had no or very few neighbours to work with). After the time elapses or the participant signals her satisfaction, KeepUP again recommends the previous 80 articles, separated into recommended and not recommended sets, using the new influence values the participant has manually defined. However, those articles that were previously rated as *interesting* or *not interesting* have their background colours set to orange and blue, respectively. Articles that are not rated have a white background. Thus, the participant is asked to simply inspect whether the recommendations have improved or degraded as a result of manipulating her neighbours' influence since the colours easily show whether KeepUP has made a mistake or not.

4.4.2 Initial Results

From a casual observation of the questionnaire feedback, there are some tentative conclusions that can be made regarding participants' satisfaction with the visualization and the ability of KeepUP to make accurate recommendations. In Table 4.1, a small selection of the questionnaire questions have been included for review. Participants are asked to give a level of agreement to each question on a 1-to-5 scale (1–"not at all [condition]"; 5–"very [condition]"). For example, the third question in Table 4.1, is about the participant's satisfaction with KeepUP's recommendations. Thus, the condition of that question is "satisfaction" and a response of 1 would mean "not at all satisfied" while 5 would mean "very satisfied." Note, not all rows sum to 13 because the first 3 participants were pilots for the study and, in some cases, their answers could not be included due to inconsistences with the setup of the study.

Overall, participants seem satisfied with KeepUP's capability to make recommendations (questions 1–3), find the service useful (question 10), and would use it if given the opportunity (question 11). There is agreement that the visualization is useful in identifying like-minded individuals (question 5) but not the tag cloud (question 6). Most participants did notice a change in their recommendations after adjusting their neighbours' influence, and 7 report that this difference was mostly positive while 3 report that it was mostly negative (not shown in the table as the answer of "mostly negative" or "mostly postive" is not consistent with the other, numeric responses). Another response of interest is that most participants report a high degree of comfort with letting others know how they rate articles (question 4) and which tags they use (not included but similar response). Finally, participants vary greatly in whether having a high level of influence on their neighbours is important to them or not. Over, the initial qualitative feedback is encouraging and shows users are interested in a social approach to filtering and reading online news articles.

4.5 Conclusions

I believe KeepUP can be applied to many different areas, from large organizations to e-learning systems. Workers need to keep up with large volumes of information and updates to documents, etc. can be easily represented by RSS feeds. Students also need to keep up with a large volumes information when considering multiple classes. In many instances it is easy to explicitly describe a relationship (student to teacher, worker to manager, etc.) and this information can be used by KeepUP in the distribution and recommendation of items. Also, there are many "hooks" in KeepUP for more powerful techniques. For instance, KeepUP uses a very simple user profile (Section 4.2.1) and it would stand to benefit from more advanced user

Table 4.1: Partial feedback (1–not all **effective**, etc.; 5–very **effective**, etc.).

	Response Counts				
Question	**1**	**2**	**3**	**4**	**5**
1) How **effective** do you believe KeepUP is at learning your preferences?	0	1	4	6	1
2) How would you rate your **satisfaction** with KeepUP's recommendations?	0	1	4	5	2
3) How **trusting** are you of KeepUP's ability to make correct recommendations to you?	0	2	5	4	1
4) How **comfortable** are you with others knowing what articles you like and don't like?	0	0	2	5	6
5) How **useful** is the visualization in identifying others who are like-minded to you?	0	1	5	5	2
6) How **useful** is the tag cloud in identifying others who are like-minded to you?	0	5	1	4	2
7) How **interested** are you in seeing others who are like-minded to you regarding specific topics?	1	0	3	3	6
8) How much **difference** in your recommendations did you notice after changing your neighbours' influence?	0	0	6	4	0
9) Would having a high level of influence on your neighbours be **important** to you?	4	3	2	4	0
10) Do you see the service that KeepUP provides as being **useful** to you?	0	0	0	8	5
11) How often would you **use** KeepUP, given the opportunity?	0	0	2	8	3

modeling techniques that could help calculate more precise activation thresholds, etc. Overall, I believe KeepUP is a solid first step to a more "social" recommender system and uses a model of how information spreads in real social networks to distribute and recommend items.

Chapter 5
Conclusions & Future Work

TIME Magazine named "You" as the 2006 Person of the Year (Grossman, 2006). The honorific is usually reserved for individuals who have the most impact on the world–good or bad. Naming "You" to the title recognizes that online community and collaboration have become an instrumental force. There is no need to look for a single exceptional man or woman amidst the traditional conflicts, failures and triumphs of the past year as the article states: "But look at 2006 through a different lens and you'll see another story, one that isn't about conflict or great men. It's a story about community and collaboration on a scale never seen before. It's about the cosmic compendium of knowledge Wikipedia and the million-channel people's network YouTube and the online metropolis MySpace. It's about the many wresting power from the few and helping one another for nothing and how that will not only change the world, but also change the way the world changes." Online communities started with a handful of technological pioneers and have blossomed into global systems that are changing how the world is perceived and how it works. The article cautions against "romanticizing" the trend, but it cannot be denied that our bustling online interactions are being felt in the real world: it is not a novel amusement or fashionable fad that is going to wind down sooner or later.

New and changing web technology, of course, has been instrumental in the community revolution. The technical and financial requirements needed to create and publish online content have been steadily declining for years. "Push-button" publishing has enabled the doubling of new blogs every five and a half months[1] and

[1]see http://www.sifry.com/alerts/archives/000419.html

the YouTube[2] juggernaut was born from advancements in video compression and streaming, among other things. As a result, online communities have changed as well, and it is more difficult to define the boundary and scope of a community–a concept that was already fuzzy. There are now many web applications that support different types of communication and different types of information storage and retrieval, but they are increasingly becoming more interconnected in the way they can exchange and combine information, e.g.: web services (c.f. Alonso et al., 2004). When a large number of people interact within a single system, it is difficult to speak about a specific online community. When interaction spans a group of systems for multiple purposes, it is nearly impossible. Also, in these large systems like YouTube, etc., there is a tendency to emphasize the discovery and delivery of content (e.g. videos, photos, etc.) over the social capital in the community.

Chapter 2 showed that it is beneficial for a community to capture interactions between its users, model the relations, and feedback the relations to its users. It is not important that any specific interpretation is attached to these relations because users can do it themselves (especially when they may be interacting for more than one purpose), but it is important for users to know that they are "bumping into" more people than they are typically aware of. That is, as a result of interaction between others and information items, users form and maintain an implicit social network. Chapter 2 described how this social network can be revealed to users and approached the engendering of this awareness from the perspective of motivating participation, but, from my review of social visualizations (Section 1.2), it is also useful to see the visualization from the perspective of helping users organize and coordinate their interactions. The results of the study carried out with a discussion forum with and without the social visualization showed that relationships were stronger between *core* and *peripheral* users in a discussion forum that included a visualization creating awareness of their relationships in comparison to the users who used the same discussion forum but did not see the social visualization.

Chapter 3 showed how social networks can be exploited to generate recommen-

[2]http://www.youtube.com

dations of information items. Through a simulation using existing movie ratings, I showed it was possible to make "good" recommendations using a graph-based approach that I called *push-poll*. This is a critical function as the amount of noise being created by online communities is significant and the task of locating anything of personal relevance becomes overwhelming. Chapter 4 explored one possible system, KeepUP, that implements the concepts of Chapter 2 and 3 together. Figure 5.1 depicts the overall process involved with my work in modeling and using social networks within online communities and is summarized as follows:

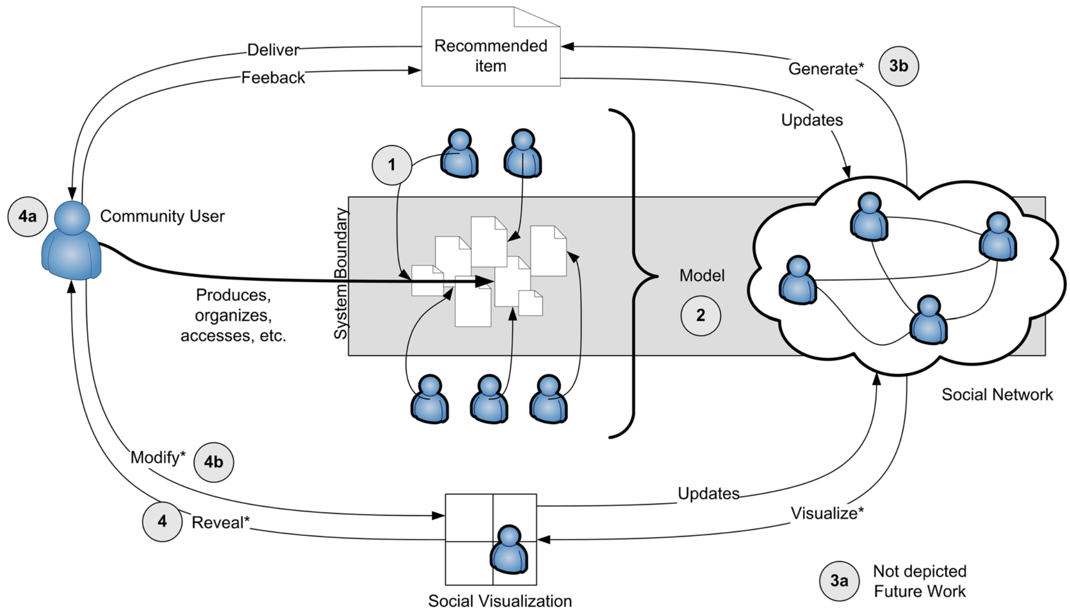


Figure 5.1: Modeling and using social networks (asterisks denote the contributions made in this thesis).

1. Users are good at aligning their actions with others, interpreting the behaviour of others, and so forth (i.e. social processes). This is represented in Figure 5.1 by the arrows that crossover the system boundary from users to items. For example, after a period of time, users' choice in tags in a collaborative tagging system will reflect the community's vocabulary (Sen et al., 2006). That is, if users label items with *IT* and *informationtechnology* to mean the same thing, then one tag will eventually become the stable "standard" as a consensus emerges. Of course, users require enough "cues" about others and their

actions within the system to achieve this type of self-organization, but the cues can be quite simple: listing only the 5 most re-occuring tags attached to an item, displaying the number of contributions each user has made when listing names of users, etc.

2. The system captures and models social processes as a social network (see Section 1.1 for a review). In Chapter 2 the social network is modeled based on the *interaction* relationships between users while in Chapter 3 it was modeled on *similiarity* relationships.

3. The system exploits the model to

   (a) identify and support social processes. This is a topic that Chapter 4 briefly touches on with the identification and support of "communities within communities." The intention is to help users "fluidly" organize into groups based on specific interests that were represented by a set of RSS feeds or keywords. This aspect is not depicted in the above figure and I leave this as future work.

   (b) generate better recommendations, etc. Chapter 3 examined how information items could be distributed and recommended using social networks. However, the overall purpose was not necessarily to make more *accurate* recommendations but to more closely follow "word of mouth" and set the foundation needed to implement Step 4b.

4. The system represents the model as a visualization and feeds it back to users who can

   (a) reflect on their actions and change their behaviour (in a manner that is preferred by the community's developers). This was the goal of the Relavis presented in Chapter 2: lurkers would be more inclined to participate if they saw that their access of information items did indeed affect the community and active contributors would be more inclined to directly engage their silent audience if they are aware of it.

(b) directly modify and improve the model to positively impact Step 3. This was the goal of the neighbour visualization presented in Chapter 4: users would be more trusting of recommendations if they had discernible involvement in helping the system generate these recommendations. Also, the presence and strength of a relationship had direct bearing on how the system performs information filtering tasks for the current users. Therefore, users have more incentive to perform expressive actions (i.e. maintain or seek out new relationships to continue receiving good recommendations). A general criticism from Relavis users was that they saw little "point" in paying attention to the relationships.

As the figure shows, recommendations and the social visualization seem to exist outside the domain of the system. It is worthwhile to point out the differences in modeling relationships *between users* and that of modeling relationships *between users and documents* (i.e. trying to understand what occurs at the system boundary between users and documents). A good example of the latter is *Intelligent Tutoring Systems* (ITS) which attempt to contextualize the user's learning through her interactions with documents, quizzes, etc. and then aid in her learning. This involves a great deal of *user modeling* and the task of building and maintaining a representative model of someone's learning behaviour, etc. is *extremely* challenging. Chapters 3 and 4 skirt the issue of user modeling for the most part and use only the most basic of models (e.g. keyword vectors). Although this is not to say that KeepUP would not benefit from a more refined and powerful user model representation, it is what can be accomplished without advanced modeling of individual users that is an interesting subject of research. The issue is where the "intelligence" occurs. Within an ITS, it is obvious that the system is responsible for intelligently predicting and fulfilling users' needs. However, when the system incorrectly interprets a user's behaviour and makes the wrong recommendation, the user's confidence in the system's future performance may be significantly reduced.

It appears some of these drawbacks are reduced in systems where interpretation occurs outside of the system. For example, with collaborative tagging systems, the system does not assign or attempt to interpret the semantics of the tags. Statistical methods are simply used to cluster items together when users search by more than one tag, and it is they who make the interesting connections between tags. In an online community, it is users who create information; users who negotiate the meaning of information; users who ultimately decide if anything is worth doing. Automated systems tend to get in the way of these aspects. My approach puts users at the heart of what really powers the system and makes it interesting. For example, digital photos represent a type of information item that is years away from automatic classification, but the Flickr community can collaboratively classify photos with just a few keywords/tags in an efficient and even *creative* manner. Part of Flickr's attraction is how it involves users to find ways in which they relate with each other through information. Of course, there is a danger that the stupidity of crowds will be harnessed more than their wisdom but, returning to the TIME article, this is what makes online communities so exciting: "this is an opportunity to build a new kind of international understanding, not politician to politician, great man to great man, but citizen to citizen, person to person. It's a chance for people to look at a computer screen and really, genuinely wonder who's out there looking back at them."

5.1 Future Work

Outside of completing the study outlined in Section 4.4, there are still a number of avenues I wish to explore with KeepUP. I believe that people recognize there more valuable information can be gained from interacting with individuals in an online community over what can be uncovered in a series of increasingly refined search engine queries. However, search engine queries are fast and usually produce good, immediate results, while becoming a part of an online community takes a greater investment of time and effort. Therefore, I would like to fully explore how to efficiently satisfy users' information needs in such a way that quickly and "organically" entangles them within the context of a community.

To that end, I believe a RSS recommender such a KeepUP is a good platform to work with. Intelligent information filtering is crucial because there are only so many feeds and items a single person can keep up with, and there are endless opportunities for community development and involvement, specifically discussing and collaborating on items that can range from the updates made on a document within an organization to a video someone has secretly captured showing abuses of political power. Lamentably, KeepUP does not yet have such discussion features, but it would be interesting to see if a discussion system can follow the push-poll approach. For example, a "discussion" would not be contained within the traditional notion of a "thread" but instead be restricted to certain areas of the social network. As the discussion grew, more users would become involved and begin pushing (i.e. recommending) it to others users in the social network. When considering systems the size of MySpace (that conveniently already have an explicit social network), this would help users gain awareness of what is outside their network's *horizon of observability* which Friedkin (1983) claims to start at two social ties away. To help keep track of discussions or find ones of interest, it would be appropriate to again apply an interactive visualization that users can use to navigate or manipulate the underlying social network.

Another topic that has already been mentioned is investigating ways of identifying and supporting communities within communities. KeepUP has taken a step in this direction with the sharing of channels (see Section 4.3); however, channels must be explicitly defined by users. I would like to implement automatic channel detection and recommendation. This would greatly decrease user effort because a list of RSS feeds can be exchanged in an *Outline Processor Markup Language* (OPML) file that is already supported by many popular RSS readers. Thus, users would only have to submit a file that can be automatically generated from their favourite RSS reader. The list of RSS feeds would then be used to recommend a set of channels (i.e. communities) that have already evolved from the feedback and interaction of existing users.

It is clear that KeepUP can be taken in many directions and the modeling of *influence* relationships and related research has only been briefly discussed. As online communities evolve and develop new methods of collaboration and coordination, it is important that they are able to attract and retain users. My current and future work in modeling and visualizing social networks is a practical and promising approach to enhancing social capital within online communities.

References

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749.

Aggarwal, C., Wolf, J., Wu, K.-L., & Yu, P. (1999). Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data-Mining*, (pp. 201–212).

Aimeur, E., & Mani-Onana, F. (2006). Better control on recommender systems. In *IEEE Joint Conference on E-Commerce Technology (CEC'06)*, (pp. 297–306).

Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004). *Web Services: Concepts, Architectures and Applications*. Data-Centric Systems and Applications. Springer.

Beenen, G., Ling, K., Wang, X., Chang, K., Frankowski, D., Resnick, P., & Kraut, R. E. (2004). Using social psychology to motivate contributions to online communities. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, (pp. 212–221). ACM Press.

Boyd, D. M. (2004). Friendster and publicly articulated social networking. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, (pp. 1279–1282). ACM Press.

Brin, S., & Page, L. (1998). Anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference*, (pp. 107–117).

Brooks, C., Panesar, R., & Greer, J. (2006). Awareness and collaboration in the ihelp courses content management system. In *First European Conference on Technology Enhanced Learning (EC-TEL 2006)*, (pp. 34–44). Springer LNCS 4227.

Bull, S., Brna, P., & Dimitrova, V. (2003). Learner modelling for reflection (lemore). Online http://www.eee.bham.ac.uk/bull/lemore/.

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

Burke, R., Hammond, K., & Young, B. (1997). The findme approach to assisted browsing. *Journal of IEEE Expert*, *12*(4), 32–40.

Burkhardt, M. E., & Brass, D. J. (1990). Changing patterns or patterns of change: The effects of a change in technology on social network structure and power. *Administrative Science Quarterly*, *35*(1), 104–127.

Bush, A. A., & Tiwana, A. (2005). Designing sticky knowledge networks. *Communications of the ACM*, *48*, 66–71.

Cheng, R., & Vassileva, J. (2005). Adaptive reward mechanism for sustainable online learning community. In *Proceedings of Artificial Intelligence in Education (AIED'2005)*, (pp. 152–159). IOS Press.

Deci, E. L., Koestner, R., & Ryan, R. M. (1999). A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin*, *125*(6), 627–688.

Dholakia, U. M., Bagozzi, R. P., & Pearo, L. K. (2004). A social influence model of consumer participation in network- and small-group-based virtual communities. *International Journal of Research in Marketing*, *21*(3), 241–263.

Dieberger, A., Dourish, P., Hook, K., Resnick, P., & Wexelblat, A. (2000). Social navigation: techniques for building more usable systems. *Interactions*, *7*(6), 36–45.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 57–66). ACM Press.

Donath, J. (2002). Visualizing online conversations. *Communications of the ACM*, *45*(4), 45–49.

Erickson, T. (2003). Designing visualizations of social activity: Six claims. In *Proceedings of CHI'2003: Extended Abstracts*, (pp. 846–847). ACM Press.

Erickson, T., Smith, D., Kellogg, M., W.A. adn Laff, Richards, J., & Bradner, E. (1999). Socially translucent systems: Social proxies, persistent conversation, and the design of babble. In *Proceedings of CHI99-Human Factors in Computing Systems*, (pp. 72–79). ACM Press.

Espinoza, V. (1999). *Social Networks Among the Urban Poor: Inequality and the Integration in a Latin American City*, chap. 4, (p. 147). Networks in the Global Village: Life in Contemporary Communities. Westview Press.

Friedkin, N. (1983). Horizons of observability and limits of informal control in organizations. *Social Forces*, *62*(1), 54–77.

Golbeck, J. (2006). Generating predictive movie recommendations from trust in social networks. In *Proceedings of the 4th International Conference on Trust Management*. Springer.

Goldenberg, J., Libai, B., & Muller, E. (2001). Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, *12*(3), 209–221.

Golder, S. A., & Huberman, B. A. (2006). Usage patterns of collaborative tagging systems. *Journal of Information Science*, *32*(2), 198–205.

Granovetter, M. (1973). The strength of weak ties. *American Journal of Sociology*, *78*(6), 1360–1380.

Granovetter, M. (1978). Threshold models of collective behavior. *The American Journal of Sociology*, *83*(6), 1420–1443.

Grossman, L. (2006). Time's person of the year: You. *TIME Magazine*. December. Online http://www.time.com/time/magazine/article/0,9171,1569514,00.html.

GroupLens (2003). Grouplens home page. Online http://www.grouplens.org.

Grudin, J. (1994). Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, *37*(1), 92–105.

Gruhl, D., Guha, R., Liben-Nowell, D., & Tomkins, A. (2004). Information diffusion through blogspace. In *Proceedings of the 13th International Conference on World Wide Web*, (pp. 491–501). ACM Press.

Herlocker, J. L., Konstan, J., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the ACM Conference on Computer Supported Collaborative Work*, (pp. 241–250). ACM Press.

Hogg, M. A., & Tindale, R. S. (Eds.) (2001). *Blackwell Handbook of Social Psychology: Group Processes*. Blackwell Publishers.

Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transaction on Information Systems*, *22*(1), 116–142.

Johnson, C. M. (2001). A survey of current research on online communities of practice. *The Internet and Higher Education*, *4*(1), 45–60.

Kautz, H., Selman, B., & Shah, M. (1997). Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, *40*(3), 63–65.

Kempe, D., Kleinberg, J. M., & Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 137–146). ACM Press.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, *46*(5), 604–632.

Kraut, R. E. (2003). *Applying Social Psychology Theory to the Problems of Group Work*, (pp. 326–356). HCI Models, Theories and Frameworks: Toward a Multidisciplinary Science. Morgan Kaufmann.

Lin, N. (1999). Building a network theory of social capital. *Connections*, *22*(1), 28–51.

Linden, G. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80.

Lock, Z., & Kudenko, D. (2006). Interactions between stereotypes. In *Adaptive Hypermedia and Adaptive Web-based Systems*, (pp. 172–181). Springer LNCS 4018.

Maloney-Krichmar, D., & Preece, J. (2005). A multilevel analysis of sociability, usability, and community dynamics in an online health community. *ACM Transactions on Computer-Human Interactions*, *12*(2), 201–232.

Mirza, B. J., Keller, B. J., & Ramakrishnan, N. (2003). Studying recommendation algorithms by graph analysis. *Journal of Intelligent Information Systems*, *20*(2), 131–160.

Nardi, B. A., Whittaker, S., Isaacs, E., Creech, M., Johnson, J., & Hainsworth, J. (2004). Intergrating communication and information through contact map. *Communications of the ACM*, *45*(4), 89–95.

Nichols, D. M. (1997). Implicit rating and filtering. In *Proceedings of the 5th DELOS Workshop on Filtering & Collaborative Filtering*, (pp. 31–36).

Nonnecke, B., & Preece, J. (2000). Lurker demographics: Counting the silent. In *Proceedings of Human Factors in Computing Systems (CHI 2000)*, (pp. 73–80). ACM Press.

Paliouras, G., Karkaletsis, V., Paptheodorou, C., & Spyropoulous (1999). Exploiting learning techniques for the acquisition of user stereotypes and communities. In *Proceedings of the 7th International Conference on User Modeling (UM'99)*, (pp. 169–178). Springer.

Perugini, S., M., G., & Fox, E. (2004). Recommender systems research: A connection-centric survey. *Journal of Intelligent Information Systems*, *23*(2), 107–143.

Preece, J. (2001). Sociability and usability in online communities: Determining and measuring success. *Behaviour & Information Technology*, *20*(5), 347–356.

Preece, J., & Maloney-Krichmar, D. (2003). *Online Communities: Focusing on Sociability and Usability*, (pp. 596–620). Handbook of Human-Computer Interaction. Lawrence Erlbaum Associates, Inc.

Preece, J., Nonnecke, B., & Andrews, D. (2004). The top five reasons for lurking: Improving community experiences for everyone. *Computers in Human Behavior*, *20*(2), 201–223.

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, *40*(3), 56–58.

Rheingold, H. (1993). *The Virtual Community: Homesteading on the Electronic Frontier*. Addison-Wesley.

Rogers, E. (2003). *Diffusion of Innovations*. Free Press, New York, 5 ed.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on E-Commerce*, (pp. 158–167). ACM Press.

Schafer, B., Konstan, J., & Riedl, J. (1999). Recommender systems in e-commerce. In *Proceedings of the 1st ACM Confernce on E-Commerce*, (pp. 158–166). ACM Press.

Sen, S., Lam, S., Rashid, A., Cosley, D., Frankowski, D., Osterhouse, F., J. nad Harper, & Riedl, J. (2006). tagging, communities, vocabulary, evolution. In *Proceedings of Computer Supported Collaborative Work (CSCW06)*, (pp. 181–190). ACM Press.

Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating word of mouth. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'95)*, (pp. 210–217). ACM Press.

Smith, M., & Kollock, P. (1999). *Communities in Cyberspace*. London: Routledge.

Sun, L., & Vassileva, J. (2006). Social visualization encouraging user participation in online communities. In *Groupware: Design, Implementation, and Use, Proceedings of CRIWG'2006*, (pp. 349–363). Springer LNCS 4154.

Takahashi, M., Fujimoto, M., & Yamasaki, N. (2003). The active lurker: Influence of an in-house online community on its outside environment. In *Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work*, (pp. 1–10). ACM Press.

Terveen, L., & Hill, W. (2001). *Beyond Recommender Systems: Helping People Help Each Other*. HCI In The New Millenium. Addison-Wesley.

Tyler, J. R., Wilkinson, D. M., & Huberman, B. A. (2005). E-mail as spectroscopy: Automated discovery of community structure within organizations. *The Information Society*, *21*(2), 143–153.

Valente, T. W. (2005). *Network Models and Methods for Studying the Diffusion of Innovations*, (pp. 98–116). Models and Methods in Social Network Analysis. Cambridge University Press, New York.

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

Webster, A., & Vassileva, J. (2006). Visualizing personal relations in online communities. In *Adaptive Hypermedia and Adaptive Web-based Systems*, (pp. 223–233). Springer LNCS 4018.

Webster, A., & Vassileva, J. (2007). Push-poll recommender system: Supporting word of mouth. In *User Modeling 2007*, (pp. 278–287). Springer LNAI 4511.

Wellman, B., Salaff, J., Dimitrova, D., Garton, L., Gulia, M., & Haythornthwaite, C. (1996). Computer networks as social networks: Collaborative work, telework, and virtual community. *Annual Review of Sociology*, *22*, 213–238.