Social Computing: An Intersection of Recommender Systems, Trust/Reputation Systems, and Social Networks

Mozhgan Tavakolifard, Norwegian University of Science and Technology
Kevin C. Almeroth, University of California, Santa Barbara

Abstract

Computational applications now go beyond personal computing, facilitating collaboration and social interactions. Social computing is an area of information technology concerned with the intersection of human and social studies connected by computer networks. The primary goal of this article is to provide a brief survey of three popular social computing services: recommender systems, trust/reputation systems, and social networks. We approach these services from a data representation perspective and discuss two of their main challenges: network sparsity and cold-start problems. We also present a novel graph model, which provides an abstract taxonomy and a common data representation model for the three services. We are mainly motivated by the power of graph theory in data representation and analysis for social computing services. Through this model, we believe that it becomes clearer that data from different contexts can be related such that new solutions can be explored; thus, it may provide illumination for the aforementioned problems and stimulate new research.

Computing applications and technology have evolved rapidly over the past decade with the advance of Internet and web technologies; the prevalence of computing resources and mobile devices; the accessibility of rich media content; and the resulting cultural and social changes. Computing is shifting to the edges of the network (i.e., networks are becoming more decentralized), and individual users are empowered with technology to use the Web for many purposes including engaging in social interaction, contributing their expertise, sharing content, and distributing information. Therefore, computer networks are inherently social networks, linking people, organizations, and knowledge. Social computing is a novel and emerging computing paradigm at the intersection of computer science and the social sciences that involves a multi-disciplinary approach in analyzing and modeling social behaviors on different media and platforms to produce intelligent and interactive applications and results. Social computing is usually referred as a groups of services that are carried out by groups of people through, for example, recommender systems, trust/reputation systems, social networks, peer-to-peer networks, Wikis, and online auctions.

Three essential characteristics of computational social science are connectivity (forming relations among people within a group), collaboration (modeling the way people interact), and community (grouping or clustering of people through functional similarity and spatial closeness) [1]. As social computing services become pervasive, many problems arise such as information overload and decision making problems. People are challenged to select products and reliable parties in transactions. As a solution, people seek advice from their friends or other trusted sources in *social networks* by using *trust/reputation systems* or using *recommender systems* to filter options according to their tastes. Thus, the focus of this tutorial article is on these three social computing services. We give a brief overview of the services with the focus on data representation. The data in all of these services can be represented as a graph-based model. The major problem is that these graphs are, in reality, often too sparse. As a result, it is difficult to make predictions for new users. We believe that by using information available in a variety of different contexts, it might be possible to solve the problems of scarcity and cold starts for new users. Thus, motivated by the power of graph theory in data representation and analysis of these services, we give an example of a common data representation as a graph-based model that exposes previously unexplored relationships among the various data elements.

Our example model neither emphasizes how the different algorithms for each service should work, nor the information an algorithm should use (e.g., in the case of a recommender system, it does not address whether an algorithm should rely on others' ratings, content-based features, or both). In addition, we make no claims about the results of algorithms being better or that they will be better received. By restricting its scope to exclude the actual aspect of social computing services, our framework provides a systematic and rigorous way to study these social computing services and stimulates new research directions on how to derive benefit from the interpretability among these services.

The remainder of the article is organized as follows. We

 0890-8044/12/$25.00

Social Computing Services/Technologies	Methods/Algorithms
Recommender systems (e.g., Netflix)	Content-based Collaborative Hybrid
Trust/reputation systems (e.g., eBay, Sporas, Histos, Epinions)	Summation or average Bayesian systems Fuzzy models Flow models
Social networks (e.g., FaceBook, MySpace, LinkedIn)	Node neighborhoods Ensemble of all paths

Table 1. *Examples of social computing services and methods.*

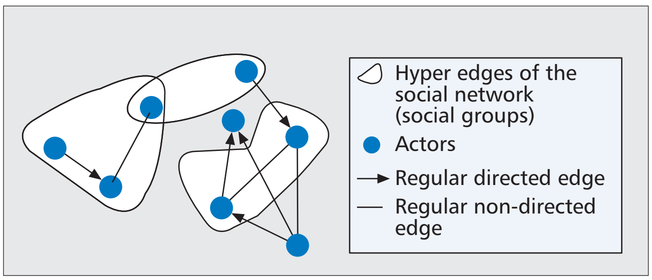

Figure 1. *Hypergraph data representation for aliation networks in social networks.*

provide a brief overview of the three social social computing services with the focus on a graph-based data representation. We explore some of the current research challenges regarding these services, specifically the sparsity and cold-start problems. Then, we give an example of a common graph model that provides for all three services. Finally, we offer concluding remarks and suggestions for future research.

Background: Overview of Social Computing Services

In this section, we present an overview of the three selected social computing services: recommender systems, trust/reputation systems, and social networks with a focus on a graph-based representation of their underlying data. Each of these services has been implemented using several methods. Table 1 shows an example of each service and some of their methods. Other work provides a thorough survey on social computing [1], recommender systems [2], trust/reputation systems [3], and social networks [4]. However, we present our brief survey with the goal of highlighting the common challenges of these services and possibility of designing a common framework as the solution. We believe that merging social networks, social trust relationships, and recommender systems can improve the accuracy of all of these services and improve a user's experience.

Social Networks

An online social network models connections among individuals or objects and facilitates information exchange between individuals or groups using relationships between users. Data is usually represented using graphs and matrices. Graph theory has been widely used to analyze social networks due to its representational capacity and simplicity [5]. In general, the properties of social network graphs have been studied extensively. However, little is known in the research community about the properties of online social network graphs at scale, the factors that shape their structure, or the ways they can be leveraged in information systems.

In social networks, the representation by graphs is also called a "sociogram," where the nodes are called actors and the edges are called relationships. The relationship can be non-directional (e.g., marriage) or directional (e.g., seller-buyer relationship). Characterizing the relationships that exist between a person's social group and his/her personal behavior has been a long standing goal of social network analysis.

Social networks are also known to be globally sparse and locally dense [4]. Given a snapshot of a social network, inferring which new interactions among its members are likely to occur in the near future is formalized as a link prediction problem. The link prediction problem asks to what extent the evolution of a social network can be modeled using features intrinsic to the network itself. The link prediction problem is also related to the problem of inferring missing links from an observed network. In a number of domains, a network of interactions based on observable data is constructed and then other likely-to-exist links are inferred. All methods can be viewed as computing a measure of proximity or similarity between nodes relative to the network topology. In general, the methods are adapted from techniques used in graph theory and social network analysis; the dynamic power of graph theory lies not in its terminology but, like any other branch of mathematics, in its theorems. Two categories of link prediction methods are as follows:

- *Node neighborhood methods*: These approaches are based on the idea that two nodes are more likely to form a link in the future if their sets of neighbors have a large overlap.
- *Shortest path methods*: These methods rank two nodes by the length of their shortest path. Such a measure follows the notion that collaboration networks are "small worlds," in which individuals are related through short chains. Some of these methods refine the notion of shortest-path distance by implicitly considering the ensemble of all paths between two nodes.

A special kind of social network is called an "affiliation network," in which nodes are actors and events to which the actors belong. Affiliation networks can also be described as collections of subsets of entities. Each event describes the subset of actors who are affiliated with it, and each actor describes the subset of events to which it belongs. Viewing an affiliation network this way is fundamental to the *hypergraph* approach.

As Fig. 1 shows, a hypergraph is a generalization of a graph, where an edge can connect any number of nodes. The nodes are actors and the edges are considered as the set of events. Furthermore, in some cases, the use of simple or directed graphs to represent the complex networks does not provide a complete description of the real-world systems under investigation. For example, in a collaboration network represented as a simple graph, we cannot know three or more users linked together in the network have collaborated on the same project or not.

Recommender Systems

The objective in a recommender system is to reduce information overload and retain customers by selecting a subset of items (e.g., movies or books) from a universal set based on user preferences. In its most common form, the recommendation problem is reduced to the problem of estimating ratings for the items that have not been seen by a user. Intuitively, this estimation is usually based on the ratings given by this user to other items and possibly other information as described below. Once we can estimate ratings for the as yet unrated items, we can recommend to the user the items with

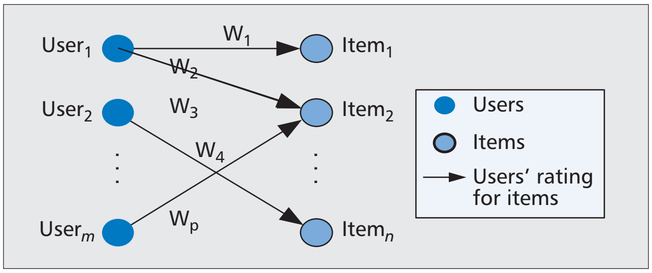

Figure 2. *Graph-based data representation for recommender systems.*

the highest estimated ratings. The new ratings of the not-yet-rated items can be estimated in many different ways using methods from machine learning, approximation theory, and various heuristics. Recommender systems are usually classified according to their approach to rating estimation and have traditionally been studied from a content-based filtering rather than collaborative design perspective [2]:

- *Content-based methods*: Similar items to the ones the user preferred in the past will be recommended to the user. In particular, various candidate items are compared with items previously rated by the user and the best matching items are recommended. For example, if a particular user reads many online articles on the topic of nanotechnology, then content-based recommendation techniques will recommend other nanotechnology articles. This recommendation will be made because these articles will have more nanotechnology-related terms (e.g., "nanooptics" and "nanobiotechnology") than articles on other topics.
- *Collaborative filtering methods*: Items that other people with similar tastes and preferences like will be recommended. For example, in a movie recommendation application, in order to recommend movies to a user, the collaborative recommender system tries to find other like-minded users (i.e., other users who have similar tastes in movies). Then only the movies that are most liked by these like-minded users are recommended.
- *Hybrid*: Several recommendation systems use a hybrid approach by combining collaborative and content-based methods. This solution helps avoid certain limitations of content-based and collaborative systems.

Current recommender systems use various kinds of data representations that usually capture three basic elements: user data (e.g., gender and address), item data (e.g., product category and price), and transaction data (e.g., user's rating, time and place of transaction). The research in recommender systems grew out of information retrieval and filtering; as a result, data is usually modeled as a user-item matrix. Another approach is a graph-theoretic model where a bipartite, directed, and weighted graph with heterogeneous nodes (i.e., users and items) and homogeneous edges (i.e., purchases) can be used to represent the data. As Fig. 2 shows, nodes represent users and items while edges represent users' ratings for items. Weights on the edges correspond to the rating values.

Despite significant research progress and growing acceptance in real-world applications, at least two major challenges limit the implementation of effective e-commerce recommendation applications. The first challenge is concerned with making recommendations based on sparse transaction data, also known as the sparsity problem. The second challenge is the lack of a unified framework to integrate multiple types of data and recommendation approaches. For better recommendation performance, a unified recommendation framework with the expressiveness to represent multiple types of input data and a generic computing mechanism to integrate different recommendation approaches is needed to fully exploit the rich information available at e-commerce sites. We explore these challenges in more detail later.

Trust/Reputation Systems

In the web, where vast amounts of content are created by users, the question of whom to trust and what information to trust has become more important and more difficult. Trust/reputation systems represent a significant trend in decision support for Internet services. The basic idea is to let parties rate each other (e.g., after completion of a transaction) and use the aggregated ratings to derive a trust or reputation score, which can assist others in deciding whether or not to transact with that party in the future [3].

Josang distinguishes between two categories of trust: *reliability trust* and *decision trust* [3]. Reliability trust is defined based on "the subjective probability by which an individual expects that another individual performs a given action on which its welfare depends." Decision trust is defined as "the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible." A *trust relationship* exists between two agents when one agent has an opinion about the other agent's trustworthiness and a *recommendation* is a communicated opinion about the trustworthiness of a third party. *Reputation* is defined as an "expectation about an agent's behavior based on information about or observations of his past actions." Therefore, reputation can be considered as a collective measure of trustworthiness (in the sense of reliability) based on the referrals or ratings from members in a community. An individual's subjective trust can be derived from a combination of received referrals and personal experience.

A reputation system uses a specific method (e.g., averaging, probabilistic-based, or belief-based) to compute reputation values for a set of objects (e.g., users, goods, or services) within a community based on the collection of recommendations from others. These reputation values may be used by the entities in the community for decision making purposes. Here, we describe some of the various methods for computing reputation and trust measures [3].

- *Simple summation or average of ratings*: The simplest form of computing reputation scores is to sum the number of positive ratings and negative ratings separately, and to keep a total score as the positive score minus the negative score (e.g., eBay) or as the average of all ratings (e.g., Epinions and Amazon).
- *Bayesian systems*: A reputation score is computed by updating probability density functions (PDFs). The updated reputation score is computed by combining the previous reputation score with the new rating.
- *Fuzzy models*: These methods represent trust and reputation as linguistically fuzzy concepts where membership functions describe to what degree an agent can be described as trustworthy or not. Fuzzy logic provides rules for reasoning with fuzzy measures of this type.
- *Flow models*: A participant's reputation increases as a function of incoming flow and decreases as a function of outgoing flow (e.g., Google's PageRank, Advogato). In the case of Google, many hyperlinks to a web page contributes to increased PageRank whereas many hyperlinks from a web page contributes to decreased PageRank for that web page.

Data for trust/reputation systems can be represented as a directed weighted graph with homogenous nodes and edges. As shown in Fig. 3, nodes are trustees and trusters (parties), edges are trust relationships, and the weights are trustworthiness values. The web of trust is often too sparse to predict trust values between non-familiar people, since in large online

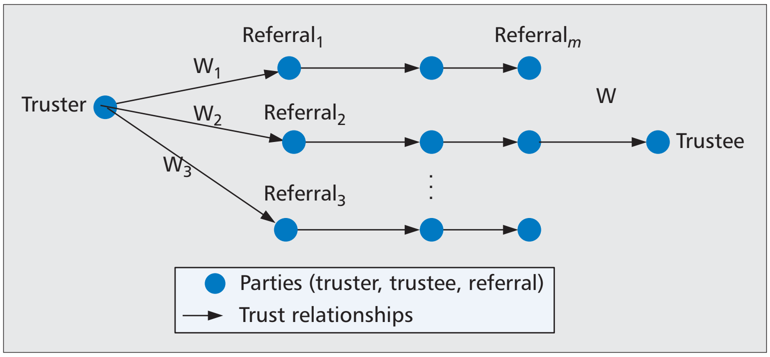

Figure 3. *Graph-based representation for trust/reputation systems.*

communities, a user has experience with only a very small fraction of the other community members. As a result, very often there will be no trust relation with an intended new partner of an e-commerce transaction.

Challenges

In this section, we provide further discussion on some of the major challenges for the three social computing services. In particular, we discuss the sparsity problem, which is one of the motivations for the graph-based representation model proposed in the next section.

The graphs in social networks, recommender systems, and trust/reputation systems are usually too *sparse*. In recommender systems, the numbers of users and items are very large. Even active users rate just a few of the total number of items; similarly, even very popular items are rated by only a few of the total number of users [2].

The *cold-start* problem emphasizes the importance of the sparsity problem. In recommender systems, this problem refers to the situation where an item cannot be recommended unless it has been rated by a substantial number of users. This problem applies to new and obscure items, and it particularly effects users with eclectic tastes. Likewise, a new user has to rate a sufficient number of items before the recommendation algorithm is able to provide reliable and accurate recommendations. In trust/reputation systems, a node must participate in interactions with others in order to raise its reputation score. As nodes in the system tend to interact with nodes with higher reputation scores, when a new node joins the system with a very low reputation score or no reputation score at all, its chance of being selected for interaction is generally rare. Hence, it is hard for a new user to raise his or her reputation score.

These problems may be alleviated by taking into consideration the interconnections among different services. As an illustration, recommendations in recommender systems are not delivered within a vacuum but rather cast within social networks. Thus, all recommender systems make connections among people either directly as a result of explicit user modeling or indirectly through the discovery of implicit relationships in data. Considering that a ratings dataset can be modeled as a bipartite graph rather than a matrix, social networks can also be formed by applying transformations on the bipartite graph, for example, two users are connected if they have rated a common item. As mentioned in the previous section, in social network theory this bipartite graph is referred to as an affiliation network.

Techniques to discover existing social networks from patterns embedded in interaction (transaction) data are analogous to collecting implicit declarations of preferences in recommender systems. Indeed, the use of social networks has expanded to many diverse application domains of recommender systems such as digital libraries and community-based service location [6].

Another example is the similarities between collaborative filtering and reputation systems. Both types of systems collect ratings from members in a community/social network. The usefulness of the former arises when the emphasis is on the content, and the latter can be used when the source of information is a more important factor. They are thus complimentary social mechanisms in global open distributed systems. There is significant potential to combine collaborative filtering and reputation systems [3].

Another example is investigating Web-based social networks and its applicability to different tasks such as trust inferrencing within trust networks. In addition, the location of a given member of a community within a social network can be used to infer some properties about his or her degree of expertise, i.e., his or her reputation [7].

However, the methods used in these examples are application-specific. This fact limits the data inputs and representations that can be used. We believe that a model should be comprehensive to support diverse inputs and representations. Furthermore, it should be flexible to support a variety of different approaches. To this end, we propose a common representation model for all three services in the next section.

In addition, for the sparsity and cold-start problems, current approaches miss many desirable aspects such as explainability of their predictions in terms and constructs that are natural to the user/application domain, effusivity and subjectivity of ratings and feedback, and coping with easy name changing. In the next section, we present an example for a joint representation graph model that facilitates the collaboration among these services.

A Common Data Representation Model

The previous section showed that the field of social computing calls for a common taxonomy, data representation, and comprehensive model. This model should have the capability to represent different types of data inputs and to support different approaches using various methods. Motivated by these needs and the analysis power of graph theory, we take a connection-oriented approach toward social computing research and suggest an example common data representation model for the three services as a solution for the sparsity and cold-start problems. Our intuition is to seek for other contextual information when the data is sparse and there is no information available for prediction. In other words, the affiliation network in social networks may be used as an underlying context for recommender systems and trust/reputation systems. As a result, by merging the graphs of all of these services, it is possible to infer missing links of one using links from the others.

Our proposed model, as shown in Fig. 4, is a heterogeneous two-layer weighted directed hypergraph in which the two layers of nodes represent users and items. Three types of links between nodes capture information about users, items, and transactions. Hyperlinks, shown as hyper edges in the figure, are social relations among users corresponding to affiliation networks in social networks. Other information about users, such as demographics, may also be added (grey edges). The links between items (dashed edges) captures the similarity between them. Different types of item information can be used to compute similarity. For products like books and movies, the product description can also be used to compute product similarity. Interlayer links (dotted edges) are formed

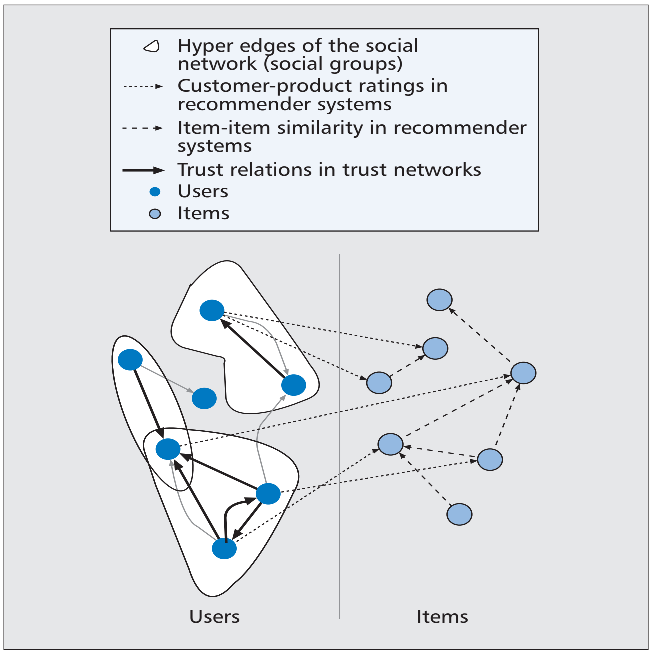

Figure 4. *A common data representation model for recommender system, trust/reputation system, and social networks.*

based on the transaction information that captures the associations among users and items (e.g., purchase history, customers' rating, or browsing behavior). Different types of transaction information may be combined in the model by assigning different weights to reflect different association strengths. For instance, a high rating on a product may be weighted higher than browsing activity, because the former reflects the user's interest more directly.

We briefly describe the use of our graph model in solving various service-related problems. Our two-layer model captures all types of data inputs and covers all the data representations that were summarized earlier. The model is flexible because different combinations of edges can be activated at run time. A rich set of analytical tools developed in graph theory (e.g., random graph search, topological graph analysis, and link prediction) can be adopted to study properties of the model such as paths and clusters that may lead to improved methods for the services. As a case in point, the recommendation problem in recommender systems can be viewed as a link prediction problem. For collaborative filtering, the non-present links (e.g., future transactions or potential interests) are predicted based on the links observed in the current graph. For the content-based and hybrid approaches, the links are predicted based on a graph that is enhanced by adding attributes about user and item nodes. In the following subsections, we explain the applicability of this model for each service in more detail.

Recommender Systems

As shown in Fig. 4, closely related users, based on their relationship in the social network (hyperlinks) or people in the same trust network (thick solid edges), are clustered into groups. Users in the same group are potential neighbors for the collaborative filtering techniques that address the sparsity problem [8, 9]. The cold start problem may also be addressed through explicit specification of a user's closest neighbors. For example, a new user joins an online bookshop. There is no information available about the previous history of book purchases by this user. However, the books purchased by his/her close friends in the social network can be used as a basis for recommendations.

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

Trust/Reputation Systems

As shown in Fig. 4, the sparsity and cold-start problems in trust networks may be improved by clustering users who are in the same social group (red hyperlinks) or users with similar historic ratings for products (dotted edges) in the same group. Then, the trust level is a common value for a group of users rather than individuals. As the groups can differ in purpose, one entity can be a member of more groups. Trust between two entities is then inferred based on their group memberships. Such models allow trust to be built between mutually unknown entities with less communication and computation load [10].

Furthermore, it is easier for the services to cope with the problem of multiple identities with this representation. In online communities, it is usually easy for members to disappear and re-register under a completely different online identity with zero or low cost. Community members can build a reputation, milk it by cheating other members, and then vanish and re-enter the community with a new identity and a clean record. In contrast, in an integrated system, it would be more costly for users to change identities in one service since they lose their current networks in the other services as well.

Social Networks

As shown in Fig. 4, a social relationship between two users may be inferred based on a mutual or a transitive trust relation between them. In this way, the existence of the trust network (thick solid edges) helps to bootstrap relations in the social network (hyperlinks) and results less sparsity.

Similar product rating patterns between two customers may also be used to induce a social relation between them. Therefore, item-item edges, which is the similarity between items in a recommender system (dashed and dotted edges), may be used to create a social relation (hyper edges) between the users who have similar ratings for those items. In the simplest form, two users are connected if they have rated a common item. The cold-start problem is less of a problem in this approach as implicit ratings bootstrap the system [6]. Perugini et al. [6] posit that recommender systems have an inherently social element and is ultimately intended to connect people either directly as a result of explicit user modeling or indirectly through the discovery of relationships implicit in extant data.

Conclusions and Future Work

In this article, we have described several challenges arising in social computing. Although these problems have been the focus of numerous papers, solutions to these problems in the context of the evolving Internet are still lacking. Specifically, in social computing, there are the problems of sparsity, cold start users, multiple identities, and context insensitivity. We have shown through a novel example how integration of three types of social computing services can help alleviate these problems.

For future work, effective solutions need to be developed for the problems identified earlier. We have briefly discussed how link prediction in one service can help to reduce the cold-start and sparsity problem in the other services. However, future researchers can look to use our example graph-based model as a basis for solving a variety of important social computing problems and investigate further how graph theory tools and techniques such as random graph search and topological graph analysis can be applied using our model to help the propagation of data and knowledge from one service into another.

References

[1] I. King, J. Li, and K. T. Chan, "A Brief Survey of Computational Approaches in Social Computing," *Proc. Int'l. Joint Conf. Neural Networks*, Atlanta, GA, June 2009, pp. 2699–2706.

[2] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 6, 2005, pp. 734–49.

[3] A. Josang, R. Ismail, and C. Boyd, "A Survey of Trust and Reputation Systems for Online Service Provision," *Decision Support Sys.*, 43, Mar. 2007, pp. 618–44.

[4] J. Scott, *Social Network Analysis: A Handbook*, Sage, 2006.

[5] C. Kadushin, *Understanding Social Networks: Theories, Concepts, and Findings*, Oxford Univ. Press, 2012.

[6] S. Perugini, M. A. Goncalves, and E. A. Fox, "Recommender Systems Research: A Connection-Centric Survey," *J. Intelligent Info. Sys.*, vol. 23, no. 2, 2004, pp. 107–43.

[7] J. Golbeck and J. Hendler, "Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-based Social Networks," *Int'l. Conf. Knowledge Engin. and Knowledge Mgmt.*, Northamptonshire, U.K., Oct. 2004, pp. 116–31.

[8] B. J. Mirza, B. J. Keller, and N. Ramakrishnan, "Studying Recommendation Algorithms by Graph Analysis," *J. Intelligent Info. Sys.*, vol. 20, no. 2, 2003, pp. 131–60.

[9] M. Papagelis, D. Plexousakis, and T. Kutsuras, "Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences," *Trust Mgmt.: Int'l Conf.*, Paris, France, May 2005, p. 224.

[10] R. Spanek, "Reputation System for Large Scale Environments," *Int'l. Conf. Digital Info. Mgmt.*, Lyon, France, Oct. 2007, pp. 621–26.

Biographies

MOZHGAN TAVAKOLIFARD (mozhgan@Q2S.ntnu.no) is a Ph.D. candidate with the Q2S Centre of Excellence at the Norwegian University of Science and Technology. Her research interests are in social networks, trust and reputation systems, recommender systems, and social computing in general. She completed her M.Sc. in information technology from Amirkabir University of Technology in February 2008 and received her B.Sc. degree in computer science and engineering from Sharif University of Technology, Iran in September 2005.

KEVIN C. ALMEROTH [SM] (almeroth@cs.ucsb.edu) is currently a professor in the Department of Computer Science at the University of California in Santa Barbara where his main research interests include computer networks and protocols, wireless networking, multicast communication, large-scale multimedia systems, and mobile applications. He has published extensively with nearly 200 journal and conference papers. He is also heavily engaged in stewardship activities for a variety of research outlets including journal editorial boards, conference steering committees, new workshops, and the IETF. He is a member of the ACM.