Recommending messages to users in social networks: a cross-site study

R. Cohen, N. Sardana, K. Rahim, D. Y. Lam, M. Li, O. Maccarthy, E. Woo, J. Zhang[†], G. Guo[†]
Cheriton School of Computer Science, University of Waterloo, Canada
[†]School of Computer Engineering, Nanyang Technological University, Singapore
{rcohen, nisardan, krahim, y7lam, m92li, omaccart, e2woo}@uwaterloo.ca {ZhangJ, gguo1}@ntu.edu.sg

***Abstract*—In this paper we produce an algorithm for presenting messages to users in social networks that integrates reasoning about the message, the author, the recipient and the social network. Our proposed model was derived on the basis of immersion within three different existing social networking environments, that of Coursera, Reddit, and medical self-help groups such as PatientsLikeMe. We first present three models, each of which is designed to perform well within the context of one specific social network. From here we derive a generalized model which can be effective regardless of social network context. We conclude with a discussion of possible directions for future research, with an emphasis on promoting the use of trust modeling and user modeling, in a view to exploring additional networks and include as well a comparison to competing models within the artificial intelligence literature. Our aim is to offer insights into coping with the massive amount of information that currently resides within our social networks.**

I. INTRODUCTION

Many different social networking environments exist currently (e.g. Facebook, Twitter, Reddit). Within each social network, the arrangements for users to view messages differ. Some networks attempt to facilitate the display of messages by predicting which messages are most valuable to show, and then ordering the presentation so that what is estimated to be more important is accessible earlier, to the user. Regardless, however, a central challenge with any of these social networking environments is the need for users to manage the massive number of potential messages to be viewed. Indeed, at times, this burden rests essentially with the user, who elects to selectively view what is available and then exits with what is hopefully the most valuable set of information.

In this paper, we examine how to design an intelligent agent that will determine which messages to show to users (and which to avoid), based a modeling of the user and a modeling of the messages, the authors of the messages and the underlying social network. Our aim is to offer a generalized model that can operate in any social network. In so doing, our aim is to be able to offer an opportunity for any existing social network to improve its ability to satisfy its participants. In addition, as new social networks emerge over time, our proposed algorithm will be able to clarify to designers how to reason effectively about which messages to show their network's users.

In order to produce our proposed algorithm, we first study in detail what happens currently with each of 3 specific social networking environments, stepping back from there to critique what is offered, to then suggest what would be best as a recommendation system within that specific social network. From there, we reason about the differences among the distinct social networking environments, to then suggest a general solution which embraces key elements of value in some of the specific environments, but that is directed to unique overall recommendations, specific to the nature of the social network in question.

After presenting our proposed solution, we will include two key points of discussion. The first is a comparison with existing models for recommendation, primarily ones that are based on trust modeling and user modeling, which are the cornerstones of our own approach. The second is reflection on valuable future directions for research, stemming from the lessons we have learned in our study of cross-site recommendation.

Our major contribution lies in identifying the central elements that should drive the recommendation of messages, regardless of social network and in clarifying which network-specific elements should also be considered, all derived on the basis of a detailed hands-on study. We highlight the theoretical big data challenge of coping with information overload and do this through our experience-based work in the area of social media.

II. SITE-SPECIFIC PROPOSALS

A. *Proposal for Coursera*

One social networking environment that is especially valuable to explore is that of Massive Open Online Courses (MOOCs): in these environments, there may be tens or hundreds of thousands of users from an extremely diverse population. As a result, information overload is a central concern. We limited our focus to Coursera and its course offering of Machine Learning, where there are forums that allow students to discuss course material.

Messages are organized in the traditional thread style format. A sidebar link from the course homepage takes you to the discussion forum. Forums are further divided into such categories as General Discussion, Programming Exercises and Course Material Feedback. By default, threads are sorted by "Last updated". There are also options to sort by "Last created" and "Top threads".

Top threads are sorted primarily by whether or not they have been pinned by course staff. Pinned threads appear before unpinned ones. Following that, threads are organised by a point system. The point system allows other users to share their general feeling toward a post without necessarily commenting back. Users can up vote or down vote a post in a thread and each post shows the net number of up/down votes beside it. Every thread is represented by: title, author of thread, author of last post in thread, timestamp of last post in thread, number of posts within the thread, number of viewers, number of points for thread and whether a member of course staff has replied.

Note that a thread gets a view when a user clicks on the link that takes them to the webpage where the thread is displayed. Visiting a thread multiple times adds to its number of views.

Top threads are currently sorted by the number of points on the first post within a thread. Most threads, however, start off with a question. Our proposed algorithm looks instead at total point activity. So, a thread with 5 upvotes on its first message and a total of 40 upvotes for all its messages would be more valuable than one with 9 upvotes on its first message and only a total of 15 upvotes for all its messages; one can optionally weight first posting votes more heavily, to integrate this influence. We found that for Assignment discussion boards, often it was within replies to messages that answers were truly offered. Our algorithm thus considers number of posts and replies to posts. We also felt that it was useful to rank Teaching Assistants (TAs) instead of simply placing more weight on the TA role. Our algorithm incorporates this as well. Our final proposal for the Algorithm (Alg.) is presented as Alg.2 with initialization outlined in Alg.1. (Note that we could imagine a variation of Alg.1 which also distinguishes the value of the TA/user for each particular type of forum. In addition, for Alg.2 we could imagine weighting some of the increments more heavily than others, e.g., to have a strong increase for the TA posting influences.)

Algorithm 1: Coursera Initialization

```
set user skill to advanced, intermediate or beginner
//can distinguish skill areas (e.g. programming, math, etc.)
// train on sample messages to determine ranking
initialize user-i-posts to 0, user-i-votes to 0 for all i
foreach message in a sample set of messages do
    if posted by user-i then increment user-i-posts by 1
    if message has upvotes then increment
    user-i-votes by total-votes
user-value = user-posts + (TA-factor * user-upvotes)
// if user is TA, set TA-factor to 2, else set to 1
sort TA array so that TA-1 has highest value,
TA 2 has second, etc.
sort user array so that user-i has highest
value, user-2 second, etc.
```

B. Proposal for Reddit

A second social networking environment selected for study is Reddit. This is an especially popular arena for connecting with peers today, one that offers a large variety of topic areas for discussion (some of which invite rather eclectic groups of users and others of which appeal to a completely heterogeneous community).

Reddit is organized into multiple communities called subreddits. Each subreddit has a specific theme or general topic which it focuses on. Some communities are default communities that each member is automatically a part of (such as movies or politics), which tend to have a more general, heterogenous audience. Non-default subreddits such as programming or history must be opted into and tend to have smaller, more focussed and homogenous communities based around their given topics.

Algorithm 2: Coursera Algorithm

```
foreach thread do
    value = total-point-activity of messages
foreach message in a thread do
    determine relevant skill area S
    if message author skill in S = recipient skill in S
    then
        increment value
    if message posted by highly ranked user then
        increment value
    if message posted by TA then
        increment value with higher increment for higher
        ranked TA
    if message has posts from multiple TAs then
        increment value
    //special case for assignment board
    if message is in Assignment board then
        increment value by total of posts and replies
    if date posted is old then
        decrement value
    if relevance to user is low then
        decrement value
    if message is first post of thread and upvotes ¿
    threshold then
        increment value
Show threads with greatest value
```

The community voting system is integral to Reddit. For each topic submission and comment that a community member (henceforth referred to as redditor) submits, other redditors may choose to vote up (positive), or vote down (negative) the given submission. Through this voting system the community decides visibility of threads and comments.

In the case of comments within a thread, highly voted up comments appear at the top, and comments that have been voted down are either hidden, or buried several levels below that must be clicked through to reach.

Reddit has a built in visibility function as a result of its voting system. Popular messages are voted up and consequently have greater visibility, and less popular posts get bubbled down and eventually hidden. This often creates an positive reinforcement loop, and biases popularity towards posts that were submitted earlier. One important goal of an algorithm that might be used to filter messages would be to give less visible, but potentially valuable messages an opportunity to surface.

Our solution builds on important observations about user preferences for messages, typical messaging characteristics and important features of the network itself. We assume a preliminary training phase where we learn the user's preference for length of posts, typicality of poster and homogeneity of the subreddit, where we anticipate increasing user preference whenever they upvote, and as well a modeling of the messages and their typicality. In fact, we have observed that some users have different preferences for lengths, for more homogenous communities, for authors who are less typical in their postings and popularity or for authors of a certain type; we include an ability to model this. We include as well some reasoning to allow posts that appear later in the thread to be presented. Our proposed algorithm for displaying messages to users is presented as Alg.3 below.

Algorithm 3: Reddit Algorithm

```
foreach message in a given community do
    if PostTypicality is typical then
        if users prefer this typicality then
            increment value else decrement value
    if ChainLength is long then
        increment value
        if messages in chain are typical then
            decrease value unless this typicality is
            preferred
    if message is topic starter then
        combine value with user preference
        if message is short to medium then
            increment value else decrement value
    else
        // message is reply
        combine value with user preference
        if message is medium to long then
            increment value else decrement value
    if Punctuation of message is poor then
        decrement value
    if Factuality is high then
        increment value
    if UserVotes is high, or similar to a poster whose
    UserVotes is high then
        increment value
    if PostTime is later than initial swarm period then
        if value is high then
            greatly increment value
    if PosterPoints is high then
        increment value
    if post conforms to bias from
    CommunityHomogeneity then
        if user prefers bias then
            increment value else decrement value
    // Final display decision
    total = value combined with MessagePoints
    if total ¿ threshold then
        display the message or promote it
    else
        hide the message or demote it
```

C. Proposal for Health Forums

A third environment that we explored for this research was that of discussion boards on the topic of health, of particular value for enabling patient-led healthcare, which has been identified as a truly critical application of the future. For this study, three different communities addressing questions of health were examined, each with varying community characteristics and purposes. Some common requirements for an algorithm to recommend messages in this environment were identified and integrated into our proposed solution.

In order to refine the scope of our discussion on healthcare social networks, we chose to focus on a common condition that affected people regardless of age, lifestyle, or ethnicity – type II diabetes. We then examined the types of messages in three social networks pertaining to self-healthcare: PatientsLikeMe, HealthTap, and eHealthForum.

The network PatientsLikeMe connects patients to other patients through a forum section where patients share their experiences; a profile section is included, used for statistical purposes. There are three ways to post a message: i) filling out a profile generated messages tagged with features such as Condition ii) filling our surveys have results propagated into a QualityOfLife message iii) simply posting on the forum. Recipients end up viewing forum-like output containing messages.

HealthTap follows a Q+A setup where doctors answer the questions of patients. Each doctor is given a trust or reputation score based both on credentials and on quality and level of contribution. The website is essentially a newsfeed with a search engine that can filter messages based on condition, symptom, etc. Recipients click on posted questions to view answers (where the message from the doctor with the highest number of likes by other doctors is displayed).

eHealthForum connects patients to doctors as well as other paitents. Messages by doctors are ordered first in a thread. Users post and view messages in a typical forum format.

Our proposed algorithm for presenting messages to users addresses some shortcomings within the current forum formats. We integrate a modeling of the reliability of a message author, the similarity of authors and recipients and the popularity of messages. The first element helps to ensure that even posters with high roles have their experience (not just their education) evaluated, the second element reflects a common occurrence where posts from dissimilar users are less valued and the third retains the important element of popular opinion. Note that reliability is intended to capture the predicted benefit of any message arriving from this particular author[1]. We propose using formulae where the relative weighting of various factors can be set, as well. The intention is for each of the factors to range in value between 0 and 1. Because whether a message is relevant to a recipient is tantamount, we propose using a function like grep to gather all pertinent messages, first. The overall algorithm (Alg.4) and its Underlying Formulas are displayed below.

III. Cross-Site Algorithm

In this subsection, we first of all step back to critique some of the existing techniques used to order the messages being presented to users in the three social networking environments examined in Section II. From here, we reach some conclusions about what should be the focus of a cross-site algorithm. We then introduce our first proposal for this algorithm.

[1]Our algorithm proposes a blend of education and experience for reliability, of value for health applications. The concept of reputation is a distinct concern, raised in our proposed general algorithm in Section III.

Health Forum Underlying Formulas:

$reliability.score = (wt.edu * edu + w.exp * exp)/(wt.edu + wt.exp)$

$similarity.score = (wt.age * age + wt.gender * gender + wt.loc * loc + w.race * race)/(wt.age + wt.gender + wt.loc + wt.race)$

$pop = 1/sqrt$(rank of messages in its thread)

$overall.quality = (wt.reliability * reliability.score + wt.similarity * similarity.score + wt.pop * pop)/(wt.reliability + wt.similarity + wt.pop)$

Algorithm 4: Health Forum Message Filter

```
Assuming good search engine algorithm or grep used
message [] getRelevantMessages(string keyword,
string userid) threads = [grep(keyword)]
for thread ∈ threads do
    popularityarray = [popularity of each message in
    thread]
for message ∈ thread do
    userposting = message.userid
    //calculate reliability score for user posting
    //assign popularity as
    //popularityarray[iterator to message]
    calculate overall quality
    if overall quality > t then
        useful.messages.append(message)
return useful.messages
```

Since thorough, hands-on investigations of the selected social networks were conducted, we emerged with some central observations about some of the better decisions made by the designers of the networks, as well as some of the shortcomings.

For the case of Coursera, we felt it was good that TA pinning raises importance of a message on messages. We noticed that there are different reasons for the different discussion boards and that there is a very diverse user base. Critiques of the existing solution include: **1)** votes on initial post within a thread used: may be problematic because have found several examples where deeper in thread was item of value; total point activity vs. being top heavy seems more representative and number of posts within thread and number of views both should dictate top thread **2)** the algorithm is agnostic to poster/author: this should be modeled **3)** the algorithm is agnostic to the relevance to the recipient **4)** while TAs should be valuable, even TAs should be distinguished by whether they are valuable or not **5)** older popular items are ranked lower but this is not necessarily best.

Considering the case of Reddit, we found that it was good to have topic filters. We observed that certain subreddits have certain message lengths typically preferred and that some subreddits are more focused/homogenous. Our critiques of the current networking solution include: **1)** users only get to see threads that are voted up highly and the main page or must use filters to see controversial things **2)** there is a bias towards earlier posts which is not always best; we need to counterbalance this **3)** length of messages seems to be a common indicator of popularity, though this is different in each subreddit and could be modeled **4)** users with high positive accounts often have desirable posts and this could be modeled **5)** long chains of replies are often of good value: this could be modeled and used **6)** individual users have preferences which are not modeled in what is bubbled up (e.g. with respect to length of chains).

Finally, for the various Health Forums that we examined, we felt that it was good that not just role but reliability (history of particular likes) were used, at times. It was also good that topics were distinguished in PatientsLikeMe, that the highest number of likes are from doctors in HealthTap and that expert responders are more valued. That some forums considered follow-up posts to be more valuable was also quite effective. It was also not unreasonable to filter out unpopular posts for all users. The critiques that emerged for us include: **1)** some messages are less relevant and still filtered to be high **2)** posts from dissimilar users were not filtered lower **3)** popularity is not modeled separately and could be, using reliability and similarity **4)** different users may have different proclivities with respect to reliability, similarity and popularity and should be seeing different solutions, but the solutions are all the same, regardless of user preferences, in the current solution.

Stepping back from all three social networking contexts, we obtain some common observations. Coursera **3)**, Health **4)** and Reddit **6)** above all point to the same general issue: solutions tend not to be user-specific; there is an absence of user modeling which, if done, could quite improve the solution. The importance of relevance to the user is another common point above per Coursera **1)** and **5)**, Health **1)**. Authors could be modeled more extensively to improve the solution per Coursera **2)**, Health **2)** and **3)**, Reddit **4)**. More can be done to be sensitive to the network itself: this is primarily drawn out in some of the Reddit items above.

The design of a cross-site algorithm should therefore be predicated on some of the following concerns: **i)** there needs to be a model of the recipient **ii)** message topics need to be modeled **iii)** author reliability needs to be modeled **iv)** there are different solutions for different networks, because there are some network-specific concerns that need to be accounted for **v)** even with a network there are some different communities, with differing characteristics **vi)** solutions need to counterbalance what the network designers have worked into the current system and what it prefers to show and in what order.

Considering Coursera, Health and Reddit in particular some of the distingushing characteristics are as follows: **i)** Coursera has a divergent user base, the importance of roles, and use of collections of messages (threads) **ii)** Reddit has a need to identify user query need, user proclivity towards typicality and the actual typicality of the network **iii)** Health forums have role importance, user/author similarity as an issue, and the need to distinguish expertise.

All of the above observations and tenets therefore lead us to the cross-site algorithm presented below as Alg.5.

The cross-site algorithm requires some modeling and some initialization as described below. Essentially, the characteristics of the social network are determined in advance. This will then decide whether certain optional features are turned on, to be modeled. The algorithm will reason with the models that are stored, optionally adjusting the decision of what messages to present by considering additional features that are modeled (or not).

Model

Four models are acquired: message, author, network and recipient. Below, an asterix indicates an optional feature (modeled if required due to the nature of the specific network). A plus flags a feature that is modeled differently, depending on the network design.

Message (Post-Time, Length, Topic, Content-Class, Votes-For+, Thread-Location*, Grammaticality*)
Author(Reputation, Role, Reliability, Similarity*, Length-Typicality*)
Recipient (Topic-of-Interest, Profile, Similarity-Pref, Grammaticality-Pref, Message-Length-Pref*, Homogeneity-Pref*, Typicality-Pref*)
Network (Role-Relevant, Thread-Based, Homogeneity*, Voter-Typicality*)

The Algorithm is initialized first as follows:

Model message, recipient, author, network

If network has roles and experts then set
If network is heterogenous or homogenous, set user proclivity
If network is thread based, algorithm is not message based/adjust
If network prefers recency, go deeper to counterbalance
/* achieved by reasoning about votes with a more complex calculation */

Set a variety of features to reflect the chosen recipient preferences
e.g. Similarity of author/recipient as important turned on or off
Set a variety of features due to initial training
e.g. Initial reputation of author

```
-------------------------------
Algorithm 5: Cross-Site Algorithm
for each message
  // come up with score and show
  // message if score > threshold
  if relevant // to topic-of-interest
    then
    if msg predicted benefit > threshold
    // based on message features
    then
    if author predicted benefit > threshold
      // based on author features
      then
      display message
-------------------------------
```

On the whole, the cross-site algorithm displayed above clarifies the potential features to consider and provides an arrangement where the features of interest are turned on or off, to be sensitive to the specific network. The algorithm integrates both user modeling and trust modeling, to ensure a proposal that is not simply generic for all possible recipients.

Initialization of the models above would be achieved in a variety of ways. For the Recipient, acquiring the values for the features is a question of selecting the appropriate user modeling approach [1]: explicit or implicit acquisition and in the case of the latter, using stereotypes or not. Most standardly, one would acquire the Profile above through some initial dialogue with the user, where values are set (e.g. age, role, etc.) and it could include a modeling of skill, best set through stereotyping (e.g. doctor skill is set high while patient skill is low). Topic-of-interest can be acquired implicitly by viewing the recipient's query (against the topic areas of the network). Similarity-preference and Grammaticality-preference have a value of No or Yes (set to No as a default) and are turned on for networks such as PatientsLikeMe. The last three features for Recipient are turned on for networks such as Reddit that are extremely diverse and that have typicality arrangements, as discussed in Section II.

Message features are initialized in part by what the network explicitly models, already. Content-Class is to track whether the message is an initial posting, a reply to a post, etc. Some features are best represented once some training has taken place. For example, length may be characterized as long, medium or short: once a series of postings have been observed, boundaries can be set to define this categorization. Grammaticality is turned-on if the user cares about this feature; the message may need to be fed to some analyzer to reveal its score for this. Thread-location is turned on for those networks where threads, not messages, are the basic units to reason about; if this is turned on, then votes-for are counted based on threads as well (e.g. combining views and votes as proposed in Section II-A).

For the Network, Role-relevance and Thread-based are known features set initially (does the network have distinguished roles and are threads the basic unit). Homogeneity and voter-typicality are only set in diverse environments such as Reddit in order to turn on these features as desirable within the Recipient model and thus to ensure that these features of the messages are also modeled. Values for these features, for messages, can be acquired as well through some training in order to determine expected values.

For the Author, role would be a defining feature, set in advance and once more stereotypical classes may be useful to employ. Reliability is here as a stand-in for basic skill level, which would first be set in expectation based on the stereotype, but which then can be adjusted over time, as one observes the behaviour of this particular individual diverging from the expectation of the class (e.g. a TA should be very knowledgeable but this particular TA seems to be less-skilled). Reputation is the most complex feature for which we would advocate using a standard trust model (e.g. [2]) in order to combine public reputation (gleaned from votes-for or popularity) and private experience. Similarity to the user, if turned-on, could be modeled through some standard metrics, as in [3], while length-typicality would need to be acquired through training.

We note that the term "message" used in this paper was intended to refer to each new posting (or reply to a posting) as a separate unit for an intelligent agent to reason about recommending to the recipient. As discussed in the context of Coursera, reasoning about entire threads turns out to be beneficial for certain social networks; as such, our cross-site algorithm leaves open the opportunity to turn on or to turn off the consideration of threads as the basic unit of recommendation.

IV. Conclusion, Related Work and Future Work

Our investigation of a cross-site algorithm for recommending messages to users in social networks, based on a concentrated study of existing social networking environments, constitutes an important step forward in the development of intelligent agents to assist users in coping with information overload in these currently popular settings for communicating with peers. Our primary conclusion is that four central elements should be modeled as part of the decision making: the messages, the authors of messages, the recipient of messages and the nature of the social network. We have also clarified which aspects of each of these elements are important to model and how best to combine a consideration of all facets in one coherent algorithm. As such, our stance is that intelligent agents designed to recommend messages should draw significantly from the area of user modeling and of trust modeling.

One trust model that is especially relevant is that of Zhang [2]. This personalized approach employs probabilistic reason to predict the benefit of agents in social networks, based on a weighted combination of private and public reputation; it can be repurposed to determine the most trustworthy messages to recommend. This research is relevant because we have determined that popularity (a kind of public reputation) is indeed a concern in many existing networks but that user preferences (personal influences) should in fact have a greater influence.

Another trust model that relates to ours is the multi-dimensional trust model of Minhas et al [4]. One important facet in this trust model is role and weighting more heavily advice provided by users with certain preferential roles, a feature we have also found important. Minhas et al. also work out in detail how best to determine majority opinion, suggesting that popularity of opinion alone is insufficient, a tenet we have drawn out in our research as well.

The research of Guo [5], [6] is perhaps most relevant to our current exploration. In particular, Guo demonstrates the value of delimiting a set of trusted neighbours for collaborative filtering recommendation [6] with a new merge method. He also clarifies how the cold start problem in recommender systems can be alleviated through certain trust modeling techniques, which again highlights the potential importance of our proposed approach of emphasizing user modeling as part of our recommendation solutions.

Other researchers have also clarified the importance of modeling the social network itself, when developing solutions for recommendation In particular, Adomavicius and Zhang [7] are able to show the importance of comparing differences between data sets from different networking environments. Franks et al. [8] draw out the influence of position within a network in order to predict the value of information to be provided by a peer. The true benefit of considering reputation within recommendation proposals, as we advocate, is also supported by research in truly collaborative social media environments, within the work of McNally et al. [9].

Our future research will also advance to explore more diverse social networks, in order to critique and expand our current cross-site algorithm. We have in fact begun a thorough investigation of Facebook. While we originally intended to showcase this network within Section II and to integrate it into our proposed cross-site algorithm, we have deferred this process until we can reflect on our findings for this context in far greater detail. This network is especially interesting because interpersonal relationships that develop influence user preferences for messages, at times at the exclusion of true topic relevance. Finally, from our study of all the networks, it was interesting to learn how different topic and community structures have evolved to assist users in managing the information that exists. To develop a truly effective overall solution for these networks requires as well a critique of how the networks have been designed and arranged, from the start. Examining Reddit's current efforts to predict upvote-downvote ratios may be of value. Other future work could assist in improving recommender system cold start efforts. For example, Gantner et al. [10] propose to learn a function to map user/item attributes to latent features of a matrix factorization model. In contrast, we have made an effort to identify which features to be modeling within any network. A final direction is to examine the scalability of our approach, through implementation; we have already begun to assemble tagged datasets.

References

[1] R. Kass and T. Finin, "Modeling the user in natural language systems," vol. 14, no. 3, pp. 5–22, 1988.

[2] J. Zhang and R. Cohen, "Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach," *Electronic Commerce Research and Applications*, pp. 330–340, 2008.

[3] N. Sardana, R. Cohen, J. Zhang, and J. Champaign, "Credibility-based trust in social networks," in *Proceedings of the AAMAS 2013 international workshop on Trust in Agent Societies (TRUST13)*, 2013, pp. 12–23.

[4] U. Minhas, J. Zhang, T. Tran, and R. Cohen, "Intelligent agents in mobile vehicular ad-hoc networks: Leveraging trust modelling based on direct experience with incentives for honesty," in *Proceedings of the 2010 IEEE/WIC/ACM international conference on Intelligent Agent Technology*, 2010, pp. 243–247.

[5] G. Guo, "Resolving data sparsity and cold start in recommender systems," in *Proceedings of the 20th international conference on User Modeling, Adaptation, and Personalization*, 2012, pp. 361–364.

[6] G. Guo, J. Zhang, and D. Thalmann, "A simple but effective method to incorporate trusted neighbors in recommender systems," in *Proceedings of the 20th international conference on User Modeling, Adaptation, and Personalization*, 2012, pp. 114–125.

[7] G. Adomavicius and J. Zhang, "Impact of data characteristics on recommender systems performance," *ACM Transactions on Management Information Systems (TMIS)*, vol. 3, 2012.

[8] H. Franks, N. Griffiths, and S. S. Anand, "Learning influence in complex social networks," in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 2013, pp. 447–454.

[9] K. McNally, M. P. O'Mahony, B. Smyth, M. Coyle, and P. Briggs, "Collaboration and reputation in social web search," in *Proceedings of the 2nd ACM RecSys'10 Workshop on Recommender Systems and the Social Web*, 2010, pp. 48–55.

[10] Z. Gantner, L. Drumond, C. Freudenthaler, S. Rendle, and L. Schmidt-Thieme, "Learning attribute-to-feature mappings for cold-start recommendations," in *Proceedings of the 2010 IEEE 10th International Conference on Data Mining*, ser. ICDM, 2010, pp. 176–185.