An Integrative Social Network and Review Content Based Recommender System

Haojie Ma and Dongsheng Che
Department of Computer Science, East Stroudsburg University, East Stroudsburg, PA 18301, USA
Email: hma@live.esu.edu; dche@esu.edu

***Abstract*—Traditional collaborative filtering (CF) recommender systems (RS) suffer the problems of poor rating accuracy, cold start and data sparsity. Friendship information is not used in current recommender systems, but such information may be useful for improving rating accuracy. Besides, traditional recommender systems only use rating scores to predict users' future buying decisions. Review comments are usually ignored, and thus it will decrease the recommendation accuracy. In this paper, a novel algorithm which integrates the social network (*i.e.* friendship information) and latent semantic analysis (*i.e.* review text) is proposed and implemented in our recommender system, aiming to improve rating accuracy. To test the performance of our recommender system, we crawled the restaurant data in the greater New York area from the website of Yelp. Our experimental results have shown that our proposed algorithm performs 18.6% higher accuracy than the traditional CF algorithm, and 1.2% higher accuracy than friend-based algorithm. We have also shown that the use of distant friendship information in the recommender systems could dramatically increase coverage rate.**

***Index Terms*—big data, collaborative filtering, machine learning, review analysis, recommender system, social network**

I. Introduction

With the development of the Internet, more and more items are sold online. Due to such overloading information, sometimes it is difficult for users to effectively find out the pieces of information that are most suitable [1] and [2]. On one hand, there are many similar items on the website, it is tricky for the user to choose the most suitable one. On the other hand, some useful items cannot be sold out due to the ignorance of the customers. Based on the above reasons, online companies developed their recommender systems (RS). RS can predict what kind of things users might be interested in buying based on their buying history. It can also recommend things to potential users based on users' similarities. Recommender system can not only help people find the most suitable products they want, but also help the e-commercial websites increase their revenues.

There are mainly two kinds of algorithms in current recommender systems: content based algorithm and collaborative filtering (CF) algorithm. Content-based algorithm makes prediction based on items properties and users profile, and then recommend the similar items which match users' profile [3]. CF algorithm assumes that users who have similar tastes in the past tend to have similar tastes in the future.

Content based algorithm works well when both items' features and users' profile information are provided. In many situations, however, it is not possible to ask the buyers to provide their full information, such as gender, age, *etc*. It is not possible to ask the seller to list full features of their goods either [4]. In the real world, CF is widely used. It has several advantages compared with the content based algorithm [5] and [6]. First, CF can make predictions without knowing the features of the items and the users' profiles. Second, CF can find user similarity or item similarity using the ratings only. However, CF suffers two fundamental problems: data sparsity and cold start problems [6]-[8].

Social network plays an important role in people's daily life, such as decision making [1] and [9], job finding, *etc*. Besides, social network can improve the accuracy of recommender system [3] and [7]. One reason is that friends share more common things than a random group of people. Evidence shows that people prefer friends' recommendation to the ones recommended by recommender system without using friendship information [10]. This is particular true in restaurants recommendation. People tend to eat with their friends. For the other reason, the integration of social network can alleviate the cold start problem. Although users don't rate any business, we can use their friends' ratings to predict the users' preferences.

In the real world, besides ratings on the business, users can also write their comments for the items they bought or used. In other words, the online reviews contain both numeric rating and plaintext feedback. Unfortunately, the review contents are always ignored by the current recommender system [11]. The reviews contain rich source of information, which is helpful for recommender system. For example, although two users give the same rating to a business or item, we have no idea of why they rate the same. While one user likes the food quality and service, the other may like the decoration and location. Through analyzing the review contents, we can discover users' preferences and business's features. Based on that, we can improve recommendation accuracy.

In this paper, social network influence information and latent semantic analysis will be integrated to improve the

Manuscript received February 5, 2015; revised May 21, 2015.

doi: 10.12720/jiii.4.1.69-75

recommendation accuracy. The reminder of the paper is organized as below: In Section II, the background will be introduced, including traditional collaborative filtering algorithm, social network based algorithm, and Latent Dirichlet Allocation algorithm. Related work will be discussed in Section III. In Section IV, we will discuss our algorithm in details. In Section V, data processing and experimental results will be discussed. Finally, conclusion will be drawn in Section VI.

II. BACKGROUND

A. Collaborative Filtering

The collaborative filtering algorithm assumes that users who share same interests on some items may also like some other items. For example, if two users (user *A* and user *B*) rate the same score on several films, user *A* may give a similar score on a new film that user *B* has rated. User-based collaborative filtering calculates pairwise user similarities based on their ratings, and then predict a user's rating on the target items using his/her top *K* nearest neighbors based on user similarities. User ratings are represented by discrete values from 1 to 5. When a user gives 1 to an item, it means that the user is very disappointed with the item. While he/she gives 5 to an item, it means that the user likes the item very much. Below is the equation of the user based collaborative filtering algorithm:

$$R_{ui} = \overline{R_u} + \frac{1}{W} \sum_{k=1}^{K} (R_{vi} * sim(u,v_k) - \overline{R_{v_k}}) \tag{1}$$

In equation (1), R_{ui} is user *u*'s predicted rating on item *i*, R_{vi} is user *v*'s rating on item *i*, R_{ui} is user *u*'s average rating in the training dataset, $\overline{R_u}$ is the average rating of user *v* in the training dataset, *sim*(*u*,*v*) is the similarity between user *u* and user *v*. *W* is the sum of the similarities of user *u*'s top *K* nearest neighbors, which can be defined as:

$$W = \sum_{k=1}^{K} sim(u,v_k) \tag{2}$$

The method calculating the similarities between user *u* and user *v* is based on Person correlation efficient [7] and [12], which is defined as below:

$$sim(u,v) = \frac{\sum_{i=1}^{N}\left(R_{ui} - \overline{R_u}\right)(R_{vi} - \overline{R_v})}{\sqrt{\sum_{i=1}^{N}\left(R_{ui} - \overline{R_u}\right)^2}\sqrt{\sum_{i=1}^{N}(R_{vi} - \overline{R_v})^2}} \tag{3}$$

In Equation (3), *N* is the number of co-related items between user *u* and *v*.

B. Social Network Based Algorithm

While current recommender system does not include social network information, one research group has investigated the usefulness of social network. Authors in [7] proposed an algorithm that combined with user preferences, item acceptance and social network. They assumed that user preference, item acceptance and friend inference are independent and use the following equation to make predictions:

$$R_{ui}' = \frac{1}{Z} \sum_{k=1}^{5} \begin{matrix} k * P(R_{ui} = k \mid A = a_i, \\ A = a_u, (R_{vi} = r_{vi} : \forall v \in U(i) \cap N(u))) \end{matrix} \tag{4}$$

Z is the normalization coefficient. The capital letter *U* is a set of users. The lower case *u* represents a user, and the lower case *i* represents an item or a business. Each user *u* in *U* has a set of preferences A_u , each item *i* in *I* has a set of attributes A_i. The value in A_u is represented as a_u , while the value in A_i is represented as a_i. $N(u)$ is the friend set of user *u*. R_{ui} is the rating user u gives to item *i*. R_{ui}' is the predicted rating of user *u* for item *i*. $U(i)$ is a set of users who have rated item *i*. *P* stands for probability.

C. Latent Dirichlet Allocation

Topic modeling algorithm is used to extract the abstract topics from the review contents. There are several algorithms for topic modeling, including latent semantic index (LSI), probability latent semantic index (PLSI), non-negative matrix factorization (NMF) and latent Dirichlet allocation (LDA). LDA was created by David Blei *et al* [13]. Currently, LDA is the most common used topic modeling algorithm, its performance and accuracy is better than LSI and PLSI [13]. So LDA algorithm will be used in this paper.

Before introducing the algorithm, we first define the variables which are used in LDA. $N(i)$ is the number of words in document *i*. *M* is the number of documents. α is the parameter of the Dirichlet prior on the per-document topic distribution. β is the parameter of the Dirichlet prior on the per-topic word distribution. θ_i is the topic distribution for document *i*. z_{ij} is the topic for the *j*th word in document *i*. w_{ij} is the j^{th} word in document *i*.

The parameters θ_i , φ_k and z_{ij} are updated by Gibbs sampling [13] and we can get the topic distribution for each document θ_i, word distribution for each topic φ_k using the following equation:

$$p(D \mid \alpha,\beta) = \prod_{i=1}^{M} \int p(\theta_i \mid \alpha)(\prod_{n=1}^{N_i} \sum_{z_{ij}} p(z_{ij} \mid \theta_i) p(w_{ij} \mid z_{ij},\beta)) d_{\theta_i} \tag{5}$$

III. RELATED WORK

Authors in [14] used network-centric recommendations. They obtained their result based on questionnaires which only contained 50 participants and with 33 people completed. The dataset was too small and large scale experiments were not carried out. In [15], authors used the trusty to filter semantic web content filtering. Actually, using trust information can remedy the data sparsity problem as seen in [16] and [17]. Their pre-condition was that they have the trust relationship between users. Such a condition requires the user in the website do extra work which may let the users escape the web. For most of the social network website, there is no such function.

This paper is similar to [7], since both use item acceptance, user preference and friendship to make predictions. But there are some differences. In [7], the authors treat all the friends equal. Friends can be divided into common friends, good friends and bosom friends. Intuitively, bosom friends can influence more than good friends while good friends can influence more than common friends. Although authors in [7] realize such problem, they simulated the situation with a few students, which was not persuasive. In this paper, latent Dirichlet allocation is used to distinguish different friends.

There are a few studies used the review content and rating together to make recommendation. In [18]-[20], authors used "Aspect discovery" to extract features from the review contents. Aspects are basically the features of products. Users can assign different weights to the features according to their preferences. For example, an artist user may assign high ratings to decoration while a user who doesn't know arts may only give a general rating to the decoration.

Authors in [19], [21] and [22] used LDA to automatically discover aspects. At the same time, they also integrated the sentiment analysis. Their goal in testing time is to predict sentiment from text while the goal in this paper is to predict user rating on the item which the user hasn't rated yet. In this paper, we not only include review text information, but also include friend information. The detailed information about our integrative recommender system in introduced as below.

IV. Proposed Approach

When people try to choose a restaurant to eat, there are a few factors that can influence their decisions. The first thing is the user's preferences. For example, if the user is a fan of spicy food, probably he will go to a Chinese Hunan restaurant instead of a restaurant selling donuts and desserts. The second factor is the attributes of the restaurant, such as location, service and *etc*. Although a user likes American food, there is a variety of American food restaurants. They provide the same kind of food, but the quality, service and location can be very different. People like the restaurants with high ratings. The last factor is friend influences. There are so many restaurants nowadays. Even the use decides to eat at an American food restaurant, and exclude some restaurants with low rating. But unfortunately, there are still a few left. If one of the user's friends has eaten in one of the left restaurants, most likely, the user will choose the one his/her friend recommends. It is called trust. Fig. 1 gives the overview of decision making process.

The probability of users' preference, item acceptance probability and friendship influences can be calculated [7]. In [7], the users having only one friend are considered for evaluation. In the real world, people can have more than one friend. Using the ratings, we can obtain both user's general rating preference, and the specific preference on different restaurants. So does the item acceptances. In this paper, we propose an Integrative Social network and Review analysis Recommender System (ISRRS) that combines friend information and Latent Dirichlet Allocation to improve predictions.

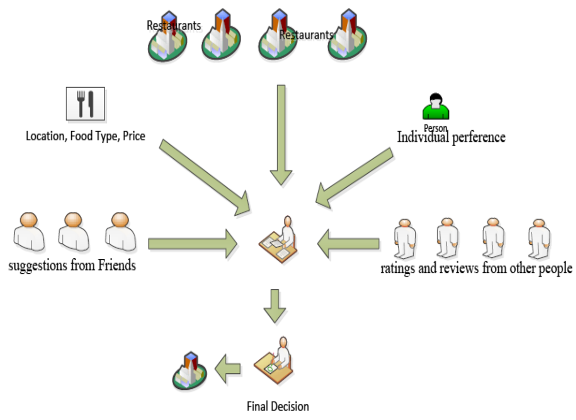


Figure 1. The decision making process in the recommender system.

Using Latent Dirichlet Allocation, document-topic distribution and topic word distribution can be calculated. All the reviews written by a user can be treated as a document while all the reviews a business receives can be also treated as a document. If all the reviews written by a user are treated as a document, the document-topic distribution is user-preferences distribution. Topic learned by LDA is the features users care most. Otherwise, if all the reviews received by a business are treated as a document, the document-topic distribution is business-feature distribution. The features of the business are the topics learned by LDA. In this paper, we will treat all the reviews written by a user as a document.

Although the information obtained from friends plays an important role in decision making, such influence is not sensitive to all the types of businesses, because people have different social circles [23], such as college circle, family circle and etc. Friends in different social circles have different influences in the related areas. For example, while the user trusts his/her friends in movies, the user may not trust them in car repairs as much as they trust them in movie recommendations.

People give different trust to different friends because they belong to different circles. It can also be applied in a single area. Even friends are in the movie circle, they still can have different tastes on movies. For example, while James shares the same interests in drama move with Kate, he also has the same taste with Tom in action movies. In this paper, after applying the LDA algorithm on the reviews contents, we can get a user-topic distribution. With the user topic distribution, the weights of different friends can be calculated by the cosine distance, which is defined as below:

$$sim(u,v) = \frac{\sum_{i=1}^{N} R_{ui} R_{vi}}{\sqrt{\sum_{i=1}^{N} (R_{ui})^2}\sqrt{\sum_{i=1}^{N} (R_{vi})^2}} \quad (6)$$

When a user has more than one friend, instead of giving the same weight to each friend, different weight can be calculated according to their preferences, and the final user influence probability can be calculated by the following equations:

$$P(R_{ui}=k\,|\,R_{vi}=r_{vi})=Z\sum_{v=1}^{V} Z_v * H(k-r_{vi}) \tag{7}$$

$$Z=\frac{\prod_{c=1}^{C} sim(u,v)*Z_v}{\sum_{h=-4}^{4}\sum_{c=1}^{C} sim(u,v)*Z_v} \tag{8}$$

In addition to the instant friends (friends directly connected), distant friends (friends not connected directly), can also influence users' buying decisions. For example, when a user wants to see a movie, he/she wants to get some advice from his/her friends, but his/her friends haven't seen the movie yet. Occasionally, one of his/her friend's friends has seen that movie, and the friend gives a high rating on the movie. After they get the information, they will watch that movie. Fig. 2 shows an example of friend relationship. Tom and Tom are immediate friends of Jim, so they are also called *one-hop friends*. Mike is Jim's *two-top friend* connected through Tom. Kate is Jim's *three-hop friend* connected through Tim and Peter. In our work, we test the influence of different friends, such as one-hop friend, two-hop friends and *etc*. The use of distant friend information could improve the coverage rate with the little sacrifice of prediction accuracy.

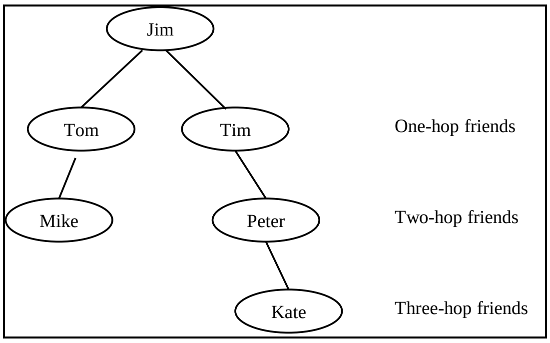


Figure 2. The friendship network showing instant friends and distant friends.

In [7], the authors use iterative classification algorithm [24] to calculate the influence of distant friends. Such kind of method treats different hop friends (*i.e.*, distant friends connected by different numbers of friends) equally. In other words, it cannot distinguish the influences between friends with different distance. In this paper, we use friends with different distance to test the assumption. We use k as a parameter to control which kind of friends can be used in the prediction. If k = 0, only the directed friends are used; If k=1, both the directed friends and the friends with 2 hops are used. By this analogy, the friends distance can be controlled.

The algorithm to get distant friends is shown in Table I. The parameter K is used to control the friends distance and α is used to control the co-rate business number of friend. Because the social network is very large, it is not possible to find all the distant friends of a user. In this paper, the threshold of friend number is set to 300. If users have more than 300 friends, we won't find any more friends for the user. According to the experiment, no more friends can be found after 5 iterations.

TABLE I. THE PSEUDO-CODE FOR FINDING DISTANT FRIENDS

```
Copy friend information to a temp list
For i =0 to K
    For each user u who has a friendList > 0
        For each friend f in the friend list
            Find the friend f's friends
            Put them into user u's friendList
            Remove user u from his friend list
        End for
        Remove duplicated friends in the friend list
        If (user u and friends co-rate businesses < α)
            Remove the friend from user u' friendList
        End if
        If (user u has more than 300 friends)
            Continue
        End if
    End for
End for
```

After obtaining user preferences, item acceptance and friend influence, we could use Naive Bayes algorithm to predict users' ratings, as shown in Equation (4).

All algorithms were implemented in Java language (JVM 1.6 and JVM 1.7). Results for Collaborative filtering algorithm were run under a 4G Memory Windows 7 machine with a 2 cores CPU. Results for Social Network Based Recommender System (SNRS), and ISRRS were run under a 66G Memory Debian machine with a 72 cores CPU. Table II lays out the pseudo code for our ISRRS algorithm.

TABLE II. THE PSEUDO-CODE FOR THE INTEGRATIVE SOCIAL NETWORK AND REVIEW ANALYSIS RECOMMENDER SYSTEM (ISRRS) ALGORITHM

```
Read Rating Info
Run Latent Dirichlet Allocation to get user-topic
distribution
Divide Rating info into 10 folders randomly
For i = 0 to 10
    Read friend Info
    Calculate user preferences
    Calculate item acceptances
    Calculate weight of each friend
    Calculate friend influences
    Predict user rating and calculate difference square
    Calculate predict coverage
    Calculate the prediction accuracy
End for
Calculate average coverage rate
Calculate average prediction accuracy
```

V. EXPERIMENTAL RESULTS

To test the performance our ISRRS algorithm, we crawled the restaurant data in the New York area from Yelp (http://www.yelp.com). XPath was used to find business profile, review information and friendship information in the webpage. Below we describe the steps used for crawling the restaurant data.

- Find all restaurants in New York City, and then extract information for each business. The information for all the business information is stored in a business list.

- For each business crawled, extract all the reviews, including review ID, business ID, user ID, review content, rating and the review date. All the review information is stored in a review list.
- Crawl user information using the user ID in the review list; the information includes user ID, name, register date and location. All user information is stored in a user list.
- Crawl the friendship information. Use the user ID which is stored in the user list to grab the friend information for each user in the user list.

By following the procedure described above, we obtained 47,319 users, 259,188 reviews and 967 businesses in the experiment; the data sparsity rate (*i.e.*, the percentage of unrated ones out of all user-item pairs in the user-item matrix) is 99.43%.

To measure the performance of recommender systems, we used two metrics: (1) Mean standard error (MSE) as defined below:

$$MSE = \sqrt{\frac{\sum_{1}^{m}\left(predict - actual\right)^2}{m}} \qquad (9)$$

where "*predict*" is the predicted rating value, "*actual*" represents the actual rating value; *m* is number of testing examples; and (2) coverage rate, which is the percentage of predicted rating number over actual rating number. In this experiment, we used a 10-fold across validation method to validate three algorithms, traditional collaborative filtering algorithm (CF), social network based algorithm (SNRS) and our integrative recommender system algorithm, ISRRS.

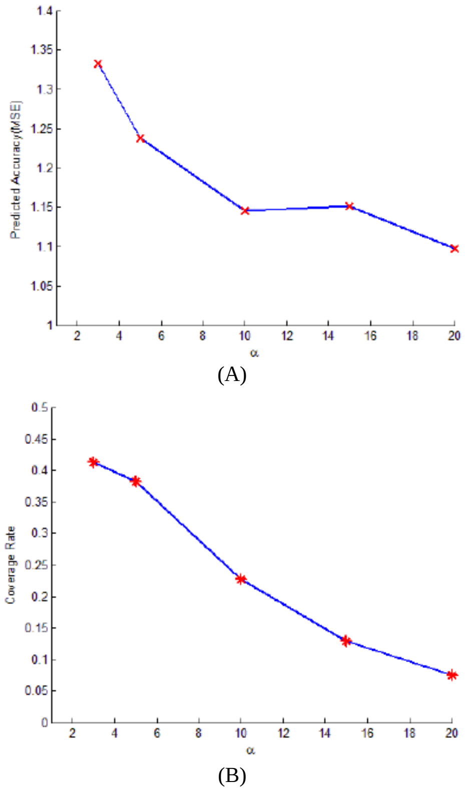


Figure 3. Performance of CF model. (A) Prediction accuracy improves as α increases; (B) Coverage decrease as α increases.

A. Performance of CF Model

In CF model, there is a parameter α which is a threshold value controlling whether users can be neighbors. In the experiment, α=1 means that two users co-rating at least one business can be treated as neighbors. Otherwise, the similarity of the two users cannot be calculated. *K* means that the top *K* most similar neighbors will be used to predict the user's rating.

As we can see from Fig. 3, with a fixed of *K* = 10, MSE starts to drop if we increase α (Fig. 3(A)). In another word, the predicted accuracy becomes higher as α increases. At the same time, the coverage rate becomes lower and lower (Fig. 3(B)). This is reasonable because when α increases, the users that are chosen have more common businesses with the target user. Thus, the similarity between the users will be more reliable. However, we cannot get large number of common businesses, thus reducing the coverage rate.

D. Performance of SNRS Model

Similar to the CF algorithm, we also used a factor α, which is used to control the number of friends, in our social network based algorithm. α can be learned based on the prediction accuracy and coverage rate. Fig. 4 shows prediction accuracy trends and coverage rates with different values of α.

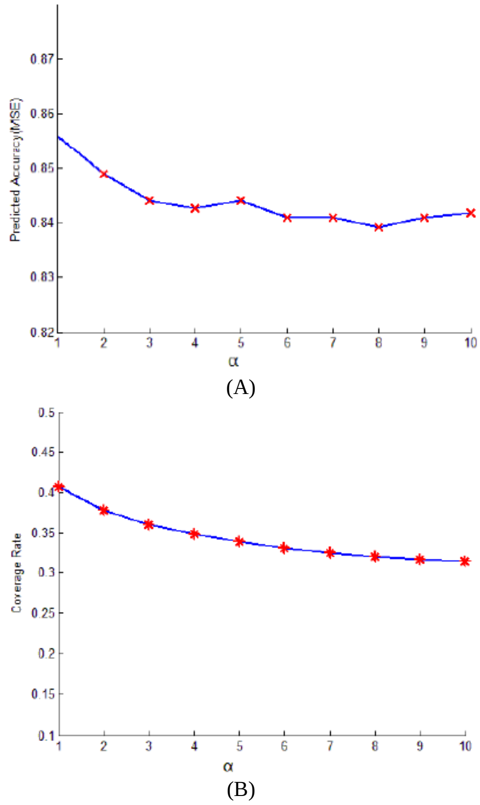


Figure 4. Performance of SNRS model. (A) Prediction accuracy improves as α increases; (B) Coverage decrease as α increases.

As we can see from Fig. 4(A), as α increases, predicted accuracy increases. This can be explained as follows: As α increases, only friends with more co-rated businesses are considered, so the predicted accuracy should increase. On the other hand, the coverage rates decrease when α increases, as shown in Fig. 4(B). This is reasonable because users who have less friends (<α) have to be excluded for evaluation, and thus decreasing the coverage rate.

As shown in Fig. 3 and Fig. 4, the use of social network data information (SNRS model) can improve the prediction accuracy over the CF model.

E. LDA Model

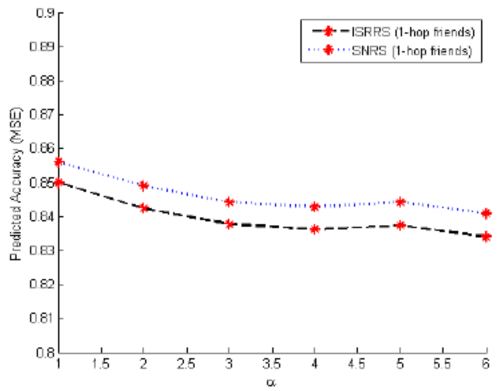
TABLE III. The Topics and Key Words Discovered Using the LDA Algorithm

Topic	Key Words
Breakfast	brunch, friend, egg, chocolate, cheese, pancake, table, wait
American	burger, frie, cheese, sandwich, meat, sauce, line, beer
Chinese	chicken, sauce, dumpling, rice, soup, noodle, pork, meat
Italian	pizza, pie, crust, slice, cheese, sauce, line, wait
Dinner	table, friend, drink, wine, night, bar, dinner, peopl
Lunch	sauce, dish, flavor, dessert, cheese, salad, bread, pork
Drinking	bar, beer, table, way, thing, peopl, night, day
Service	wine, experience, meal, course, star, dishe, dinner, table
Japanese	ramen, pork, noodle, bun, broth, sushi, wait, friend
Mexican	taco, chicken, arepa, sauce, corn, empanada, chip, cheese

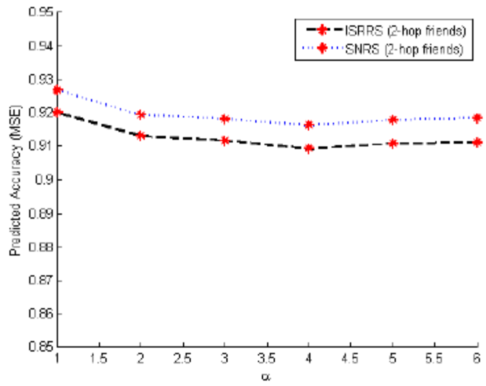
The Latent Dirichlet Allocation algorithm was implemented to find the topics discussed in the review contents. The numbers of topics for 5, 10, 20 and 50 were tested in LDA. According to the LDA algorithm, we found that number of topics of 10 best explains the content written in the reviews. As we can see from Table III, the ten topics we found are breakfast, American, Chinese, Italian, Dinner, Lunch, Drinking, Service, Japanese and Mexican. The ten topics were inferred from keywords, which are listed on the right side in Table III, where only the first 8 keywords were chosen for each topic in this experiment.

F. Performance of ISRRS Model

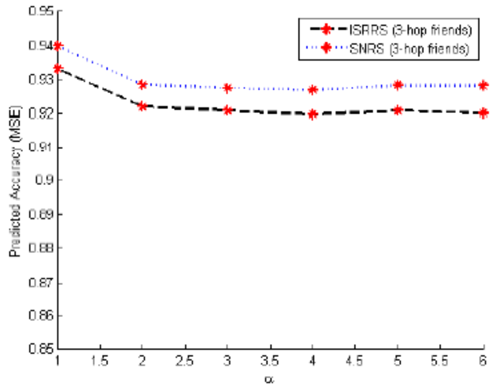
To examine whether the review content incorporated into social networked recommender system can improve prediction accuracy further, we have developed our ISRRS model.



(A)



(B)



(C)

Figure 5. Prediction accuracy comparison (based on MSE) between ISRRS and SNRS.

As we can see from Fig. 5, our ISRRS model performs better than the SNRS model consistently, regardless the use of 1-hop friends (Fig. 5(A)), 2-hop friends (Fig. 5(B)), or 3-hop friends (Fig. 5(C)), indicating the integration of review contents using LDA algorithm can improve prediction accuracy.

Another interesting finding is that: compared to the model using direct friends (1-hop friends), the models using distant friends (2-hop friends or 3-hop friends) have lower prediction accuracy in terms of MSEs. For instance, the MSE values for one-hop friend, two-hop and three-hop friends ISRRS models (α =1) are 0.83, 0.92 and 0.93 respectively. This can also be seen in the SNRS models.

While the use of distant hop friendship information may decrease prediction accuracy, the coverage rate can be dramatically improved. As shown in Table IV, the coverage for one-hop friends ISRRS model (α =6) is 33.1%, while the coverage for two-hop friends ISRRS model (α =6) is 73.8%, 122.96% increase of coverage rate.

TABLE IV. The Summary of MSE and Coverage Rate of ISRRS with Different Friend Distances

Friend Distance	MSE	Coverage
one-hop friends	0.83	33.1%
two-hop friends	0.91	73.8%
three-hop friends	0.92	77.2%
four-hop friends	0.92	78.1%
five-hop friends	0.92	78.4%

VI. Conclusion

In this paper, we have introduced our recommender system, ISRRS, which integrates social network and the review contents into traditional CF model. Compared with the traditional CF model, ISRRS performs 18.6% better in terms of MSE values. ISRRS also performs better than the state-of-art social network based recommender system model. Through using latent Dirichlet allocation algorithm, we could learn the topics from review contents, and applied such information into our integrative recommender system. Additionally, we have discovered that distant friendship information could dramatically increase coverage rate, with small reduction of prediction accuracy.

In our future work, we will apply our model into more locations and business categories. We hope improved prediction accuracy of our model could help business get more insights, and help them provide direction in order to attract more users.

Acknowledgment

This work was partially supported in part by a grant from Faculty Development and Research (FDR) Major Grant (#2014073) at East Stroudsburg University, USA.

References

[1] F. E. Walter, S. Battiston, and F. Schweitzer, "A model of a trust-based recommendation system on a social network," *Autonomous Agents and Multi-Agent Systems,* vol. 16, no. 1, pp. 57-74, 2008.

[2] R. Burke, A. Felfernig, and M. H. Göker, "Recommender systems: An overview," *AI Magazine*, vol. 32, no. 3, pp. 13-18, 2011.

[3] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," in *Proc. 4th International Conference, iTrust*, Pisa, Italy, 2006, pp. 93-104.

[4] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating 'word of mouth'," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, Denver, CO, USA, 1995, pp. 210-217.

[5] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proc. the Eighteenth National Conference on Artificial Intelligence*, 2002, pp. 187-192.

[6] N. Good, J. B. Schafer, J. A. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining collaborative filtering with personal agents for better recommendations," in *Proc. Sixteenth National Conference on Artificial Intelligence and the elEventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, 1999, pp. 439-446.

[7] J. He and W. W. Chu, "A social network-based recommender system (SNRS)," in *Data Mining for Social Network Data*, Springer, 2010, ch. 4.

[8] G. Adomavicius and A. Tuzhilin, "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,* vol. 17, no. 6, pp. 734-749, 2005.

[9] S. Battiston, E. Bonabeau, and G. Weisbuch, "Decision making dynamics in corporate boards," *Physica A: Statistical Mechanics and its Applications,* vol. 322, pp. 567-582, 2003.

[10] R. R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends," presented at DELOS workshop: personalisation and recommender systems in digital libraries, Dublin City University, Ireland, 18-20 June 2001.

[11] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proc. 7th ACM Conference on Recommender Systems*, 2013, pp. 165-172.

[12] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conference on Computer Supported Cooperative Work*, New York, NY, USA, 1994, pp. 175-186.

[13] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[14] A. Sharma and D. Cosley, "Network-centric recommendation: Personalization with and in social networks," in *Proc. IEEE Third International Conference on Privacy, Security, Risk and Trust (Passat) and 2011 IEEE Third International Conference on Social Computing (Socialcom)*, Boston, MA, 2011, pp. 282-289.

[15] J. Golbeck, "Combining provenance with trust in social networks for semantic web content filtering," in *Provenance and Annotation of Data*, Springer, 2006, pp. 101-108.

[16] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," in *Proc. Federated Int. Conference on the Move to Meaningful Internet Systems*, 2004, pp. 492-508.

[17] P. Massa and B. Bhattacharjee, "Using trust in recommender systems: An experimental analysis," in *Trust Management*, Springer, Berlin Heidelberg, 2004, pp. 221-235.

[18] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content," *WebDB*, vol. 9, pp. 1-6, 2009.

[19] I. Titov and R. T. McDonald, "A joint model of text and aspect ratings for sentiment summarization," in *Proc. ACL*, 2008, pp. 308-316.

[20] H. Wang, Y. Lu, and C. Zhai, "Latent aspect rating analysis on review text data: a rating regression approach," in *Proc. 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, 2010, pp. 783-792.

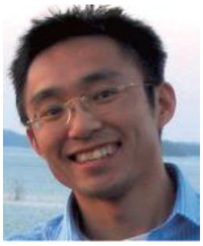
[21] Y. Jo and A. H. Oh, "Aspect and sentiment unification model for online review analysis," in *Proc. Fourth ACM International Conference on Web Search and Data Mining*, New York, 2011, pp. 815-824.

[22] B. Lu, M. Ott, C. Cardie, and B. K. Tsou, "Multi-aspect sentiment analysis with topic models," in *Proc. IEEE 11th International Conference on Data Mining Workshops*, Vancouver, BC, 2011, pp. 81-88.

[23] W. Verbeke and S. Wuyts, "Moving in social circles—social circle membership and performance implications," *Journal of Organizational Behavior*, vol. 28, no. 4, pp. 357-379, 2007.

[24] J. Neville and D. Jensen, "Iterative classification in relational data," in *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, Austin, Texas, 2000, pp. 13-20.

Haojie Ma graduated as a master student from the Department of Computer Science at East Stroudsburg University (ESU) of Pennsylvania. He received his master degree in 2014. His advisor is Dr. Dongsheng Che. His research interests include machine learning, data mining and big data analysis. He is currently working for TMG Health as an Information Management Analyst.



Dongsheng Che is an associate professor in the Department of Computer Science at East Stroudsburg University (ESU) of Pennsylvania. He received his PhD degree in computer science from The University of Georgia in 2008 and has since been on the faculty at ESU. His main research interests are in the areas of bioinformatics, machine learning, data mining, big data analysis, and evolutionary computation. He has more than 40 publications of his research in peer-reviewed journals and conferences including *Bioinformatics*, *BMC Bioinformatics*, *BMC Genomics*, *International Scholarly Research Notices*, *Nucleic Acids Research*, and *Proteins*.