

Social Commerce Hybrid Product Recommender

Rahul Hooda

M.Tech.(Software Engineering)
U.I.E.T, Kurukshetra University,
Kurukshetra, Haryana, India

Kulvinder Singh

Assistant Professor
U.I.E.T, Kurukshetra University,
Kurukshetra, Haryana, India

Sanjeev Dhawan

Assistant Professor
U.I.E.T, Kurukshetra University,
Kurukshetra, Haryana, India

ABSTRACT

The vision for Web 3.0 (also known as *Semantic Web*) is the ability to create meaning out of huge quantity of qualitative data. Existing data can be interconnected for further uses. Web 2.0 focused on the users interaction with others whereas Web 3.0 focus more on the users themselves. The advantages of Semantic Web and E-commerce give rise to social commerce (also referred as f-commerce). The future of business lies on the “social” factor and it is this factor which gives rise to a new kind of connected consumers who are becoming influential in their own right. This paper explores a very specific instance of Semantic Web – Social Recommender System. This paper discusses the likelihood of converting social data into quantitative information and using this information to power social recommendations. This paper first outlines the benefits of social commerce over e-commerce platform. Then the related literature work regarding hybrid recommenders is discussed. Next it is discussed how to predict ratings from a user-item rating network and friend’s network and then how to unify similarity matrices obtained from different networks. And lastly this paper covers the social hybrid product recommender algorithm and its experimental evaluations to predict its efficiency.

Keywords

Social commerce, hybrid product recommender, f-commerce, homophily, Cosine similarity, Smith Waterman string similarity measure, unipartite graph, bipartite graph

1. INTRODUCTION

The merger of e-commerce and social platforms is unlocking new and important connections between people, brands, content and data. This merger of e-commerce and social networks gives rise to “Social Commerce”. The term “Social Commerce” was introduced in November 2005 by yahoo, to describe how online collaborative shopping tools such as shoppisphere and pick lists can be used to provide personalized recommendations.

Consider a situation when one is recommended those products which are bought, liked or shared by their close ones. This data of what their near ones buy, like or share comes from social networking sites. Social commerce is all about fostering relationships with people and the varying groups of consumerism they represent. It is social commerce that brings the need to integrate a 5th P into marketing, that is *People*. Web 3.0 is data driven. This data in the form of relationships, buy, like, share comes from the people themselves.

Almost all popular e-commerce websites and brands like Amazon.com, Levi’s, 1-800-flowers.com, Dunkin Donuts, Starbucks, Coca-Cola, Red Bull, Cafepress, Groupon, Houzz, LivingSocial, ShopSocially, Tabjuice, OpenSky are using social commerce platform to increase their profits.

Levi’s “Friends Store” displays the jeans that the customer friends have admired and also allows the customer to share the jeans they like. Friends store is an example of peer-to-peer influence model where the customer’s affect and are affected by those they trust. Friends store creates a bridge between social web and web, relationships and content, thus socializing the objects that move us.

E-commerce startups like EasySocialShop.com, Ondango.com and Soldsie.com use Facebook based shopping platforms to leverage their profits. EasySocialShop lets sellers integrate existing shops with their Facebook pages and list their items there.

IBM WebSphere Commerce is a social commerce tool which presents a buyer-centric platform to help retailers bond and promote active participation with buyers to improve the buyer experience. On integrating the tool with Facebook, retailers and merchandisers can target buyers based on what they “share” and “like” while on Facebook.

In this paper, a hybrid recommender system which combines multiple similarity matrices derived from heterogeneous implicit (User-item rating network) and explicit social networks (Friends network) is proposed. The proposed system uses Cosine similarity to compute similarity among users of a bipartite graph and weighted sum method to compute the similarity among users of a unipartite graph. The proposed system uses a string similarity matching algorithm to match the product keywords with the keywords retrieved from the user search history. The string similarity algorithm will be used to reorder the list of recommended products, with those products first which are being searched by the user in the past. The system will adjust the influence of user-item rating network and friend’s network for each target user. For example, if a user has very few friends in the friends network and has rated many items in the user-item rating network then the weighting strategy of our system endorse the information given by implicit social network.

1.1 Why use Social Commerce

- Change the way products are brought to market: Allow customers to vote on their favourite products they want to purchase before they are introduced in stores.
- Enable peer recommendations: Those items are recommended which are bought or liked by customers who influence you, like friends and family members. Levi’s friends store provides peer recommendations.
- Increase relationships with others who share your taste: Social commerce enables customers to build relationships with other customers with whom they never met in real life.
- Develop social shopping opportunities: Enable customers to purchase on social networks like Facebook, Google Plus to increase the likelihood that they will buy. 1-800-

flowers.com was the first retailer to test shopping from Facebook’s news feed.

- Build brand awareness cost effectively: Companies use social commerce platform to increase the number of conversations about their products. This has the added benefit of increasing customer trust.

2. RELATED WORK

This section deals with the previous work related to product recommender systems. First, collaborative filtering, which have been used for recommendation in user-item networks is discussed. The GroupLens [1,2] system was developed for filtering text documents (i.e., news articles) but also introduced the basic idea of automatically finding similar users in the database for making predictions. Moreover, item-based CF [3, 4] is based on similarity between items, instead of similarity between users. NewsDude [5] presented by Billsus & Pazzani (1999), a content-based filtering system suggests new stories the user might like to read. To accomplish this two user models are built. The first user model measure similarity between the new story and the stories that the user has read before by counting the co-occurrences of words appearing in these stories. The second user model assigns a probability of interest to a new story by comparing how frequently its words occur in those stories the user regards as interesting to those the user regards as of no interest.

One way to build a hybrid recommender system is to implement content-based and collaborative recommender systems separately and then combine the ratings obtained from individual recommender systems into one final recommendation. Daily Learner system [6] presented by Billsus & Pazzani (2000), selected the recommender system that gave the recommendation with the higher level of confidence, while hybrid recommender system presented by Tran & Cohen [7] chose the one whose recommendation were more consistent with past ratings of the user. Many hybrid recommender systems, including the “collaboration via content” approach described by M. Pazzani (1999) [8] and the Fab system [9], were based on traditional collaborative techniques but also maintained the content-based profiles for each user. Fab system [9] used content-based filtering, which ranked documents and considered user’s feedback to update their personal selection agent’s profile. As presented by M. Pazzani [8], this allows solving sparsity-related problems of collaborative filtering approach. Basu *et al.* (1998) [10] proposed to use content-based (genre of movies) and collaborative characteristics (age or gender of users) in a single rule-based classifier. Popescul *et al.* [11] and Schein *et al.* [12] proposed a unified probabilistic method for combining collaborative and content-based recommendations. Several papers, such as [8,9,13,14] states that hybrid approaches provide more accurate recommendations than traditional approaches.

There are many techniques that combines information from explicit and implicit social networks. TidalTrust [15] and MoleTrust [16] combines the user-item rating data with the data obtained from explicit trust-based social networks to increase the item recommendation accuracy. TidalTrust [15] executes an adapted breadth first search in the trust network to calculate a rating prediction whereas MoleTrust [16] contemplates paths of friends to a defined depth. Jamali and Ester [17] suggested a trust propagation mechanism with matrix factorization technique for making item recommendations in social networks. He and Chu [18]

probabilistic model make item recommendations based on information in social networks, including item’s universal acceptance, user likes and influence from friends.

3. RATING PREDICTION BASED ON USER-ITEM RATING NETWORK

As described in Section 1, the proposed hybrid recommender system will use implicit social network (user-item rating network) to form a user-item similarity matrix.

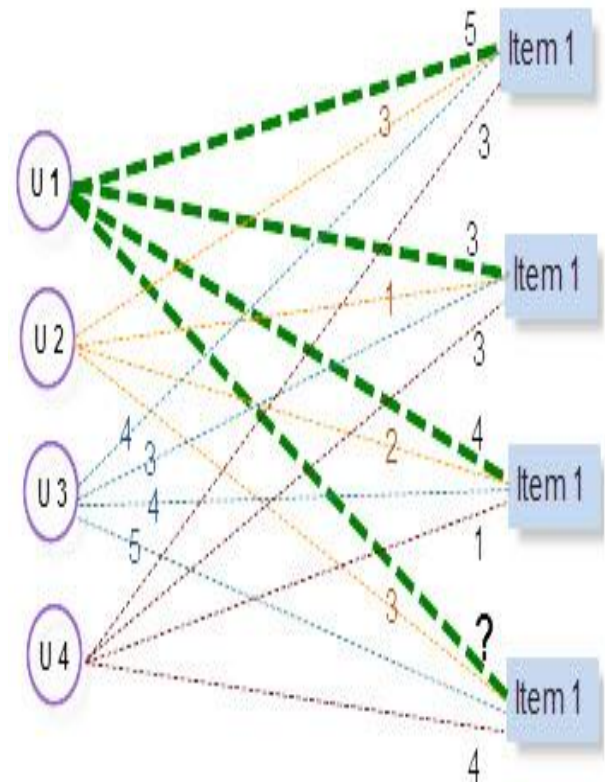


Figure 1: User-item rating network

The above shown bipartite graph, can be also shown by a matrix R, where the rating of a user u over an item i is given by the element $R(u,i)$.

Table 1: User-item rating matrix R

	Item 1	Item 2	Item 3	Item 4
U1	5	3	4	?
U2	3	1	2	3
U3	4	3	4	5
U4	3	3	1	4

Table 1 represents rating matrix, where U1-U4 are users. The possible rating values are defined on a numerical scale from 1 (strongly dislike) to 5 (strongly like). The cell with no rating is represented by a question mark. Now to calculate the rating similarity matrix sim_R , the proposed system uses collaborative filtering approach. The idea behind collaborative filtering is

simply as follows: given a user-item rating database and the ID of the current user as an input, identify other peer users that had similar rating preferences to the current user in the past. A rating prediction is computed for every item that the current user has not seen yet. The ratings computed for an item depends on the ratings given by the peer users for that item. The underlying assumptions of such methods are that (a) if users had similar tastes in the past they will have similar tastes in future and (b) user preferences remain stable and consistent over time. Related work in Collaborative Filtering has used Cosine similarity or Pearson correlation to calculate the similarity among users (sim_R). The proposed system uses cosine similarity (Equation 1) to measure the similarity between two users u and v , where $r_{u,i} = R(u,i)$.

$$sim(u, v) = \frac{\sum_{v \in I} (r_{u,i} * r_{v,i})}{\sqrt{\sum_{v \in I} (r_{u,i})^2} \sqrt{\sum_{v \in I} (r_{v,i})^2}} \quad (1)$$

Table 2: sim_R user item-rating similarity matrix

	U1	U2	U3	U4
U1	1	0.766	0.783	0.669
U2	0.766	1	0.975	0.916
U3	0.783	0.975	1	0.936
U4	0.669	0.916	0.936	1

Table 2 shows the user item-rating similarity matrix, sim_R . Now to predict the rating of user 1 on Item 4 equation 2 can be used, which is as follows:

$$pred\ rating_{u,i} = \frac{\sum_{v \in U} [sim(u, v) * r_{v,i}]}{\sum_{v \in U} sim(u, v)} \quad (2)$$

Using equation 2, the rating prediction of U1 on item 4, is equal to 4.007 [((0.766*3)+(0.783*5)+(0.669*4))/(0.766 + 0.783 + 0.669)].

Equation 1 does not consider the fact that users are different with respect to how they interpret the rating scale. Some users tend to give only high ratings, whereas others will never give a 5 to any item. To deal with this equation 2 can be modified as:

$$pred\ rating_{u,i} = avg + \frac{\sum_{v \in U} [sim(u,v) * (r_{v,i} - avg_v)]}{\sum_{v \in U} sim(u,v)} \quad (3)$$

Where i is any unrated item, avg is average rating of items by user u in rating matrix R . Also avg_v is average rating of items by user v . Corresponding terms in the previous summations of equation 3 should be deleted if some user v has not rated item i . In the above example we have considered the similarity value of every user to calculate the rating prediction of user 1 on item 4. But in real situations there will be many users (neighbors), so instead of considering similarities of all users, our proposed system will take into account the similarities of say top n users.

4. RATING PREDICTION BASED ON FRIEND'S NETWORK

A Friend's network is also known as peer to peer influential network because each user is influenced by the actions of its peer users (neighbors). Rating prediction using friends network assumes that any item which your closest friends rate will be a good recommendation for you. Friends network is subject to the principle of homophily [19], which states that a person has relationships with people who are most like him and therefore the person's preferences and his friend's preferences will be highly correlated and a person's preferences are highly motivated by those around him- so if a friend of mine like a product then I am more likely to try it. Using Friend's network the user-user similarity matrix, sim_S can be calculated easily and using the same approach (equation 2) discussed in the previous section our system will predict the rating for unrated items.

Let G be a simple unipartite graph in which there are no multiple edges between the nodes and there are no self loops on any node. The graph is undirected and the edges are weighted according to the relationship between the nodes. The weights represent the closeness (similarity) between the nodes.

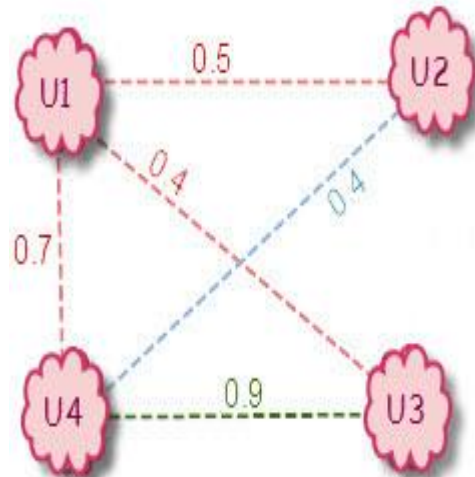


Figure 2: shows unipartite friend's network

There exists a variety of similarity measures that can be used to compute the user-user similarity matrix like Random Walk with Restart, FriendTNS, Common Neighbors index, Jaccard Coefficient, Adamic and Adar index etc. for analyzing the closeness of nodes in a network.

Table 3: shows user-user adjacency matrix S .

	U1	U2	U3	U4
U1	0	1	1	1
U2	1	0	0	1
U3	1	0	0	1
U4	1	1	1	0

To calculate the closeness between nodes, several attributes from users profile like profession, family member, college friend, school friend, birth place, current location etc can be considered. These attributes can be ordered according to their priority like profession > family member > college or school

friend and weights can be assigned to these attributes accordingly. Weights can be assigned using Rank-sum weighting formula, Rank exponent formula, Rank reciprocal formula or Rank order centroid formula. Our proposed system considered the above mentioned three attributes. We assigned “same profession” attribute a 0.4 weight, 0.3 to “family member” and 0.2 to “college or school friend”. The proposed system uses simple weighted sum formula to calculate the similarity (closeness) between nodes. Consider the diagram in figure 2, the weight on edge connecting nodes U1 and U2 shows that U1 and U2 are family members as well as college or school friends. The weight on edge connecting nodes U1 and U4 shows that U1 and U4 has same profession as well as they are family members also.

Table 4: shows user-user similarity matrix sim_s

	U1	U2	U3	U4
U1	0	0.5	0.4	0.7
U2	0.5	0	0	0.4
U3	0.4	0	0	0.9
U4	0.7	0.4	0.9	0

By considering the above user-user similarity matrix and equation 2, the rating prediction of U1 on item 4, is equal to 3.937 $(((0.5 * 3) + (0.4 * 5) + (0.7 * 4)) / (0.5 + 0.4 + 0.7))$

5. UNIFYING sim_R and sim_S INTO A SINGLE MATRIX

Both the similarity matrices can be combined into a single matrix by using equation 4 which is as follows:

$$sim(u,v) = \alpha * sim_R + (1 - \alpha) * sim_S \quad (4)$$

Similarity between users u and v can be calculated using equation 4, where α takes value between $[0,1]$.

Before unifying both the matrices into a single matrix one must consider the fact that the distribution of similarity values in the interval $[0,1]$ between sim_R and sim_S differ significantly. For example the similarity values in sim_R are distributed between 0.6 and 1, whereas the similarity values in sim_S are distributed between 0 and 0.9. So, it is unfair to take a simple weighted average of both the similarity matrices using equation 4. So the system first transform both the matrices using equation 5.

$$simq(u,v) = \frac{simq(u,v) - m}{s} \quad (5)$$

Where m is the mean similarity of matrix q and s is the standard deviation of matrix q . Then the similarity values are normalized in the interval $[0,1]$.

Symeonidis *et. al.*[20] proposed that parameter α can be auto adjusted using equation 6.

$$\alpha = \frac{dR}{dS + dR} \quad (6)$$

where $dS = (\text{local } S / \text{global } S)$ is the local to global coefficient of the selected user into the user-user adjacency matrix S and $dR = (\text{local } R / \text{global } R)$ is the local to global density coefficient of the selected user into the user-item rating matrix R . local S is the number of non-zero values in selected user row (adjacency matrix S) divided by the number of users. global S is the number of non-zero values in the full

adjacency matrix S divided by the square of number of users. local R is the number of non-zero values in selected user row (user-item rating matrix R) divided by the number of items. global R is the number of non-zero values in the full user-item rating matrix R divided by the number of users and number of items.

6. SOCIAL HYBRID PRODUCT RECOMMENDER ALGORITHM

Algorithm Social Hybrid Product Recommender

Input:

U: user for which the item ratings are to be predicted

NU_i: where $i=1$ to k be the immediate neighbors of user u along with their rating on various items.

K: set of keywords for each individual items

H: product search history

Output:

O: set of recommended products.

for $i= 1$ to k

 compute the user-item similarity matrix sim_R using equation 1

 compute the user-user similarity matrix sim_S using weighted sum formula

end for

transform both the similarity matrices sim_R and sim_S using equation 5 and then scale them in the interval $[0,1]$

compute the local and global densities i.e., local S , global S , local R and global R

calculate adjustment parameter α using equation 6

unify both the similarity matrices sim_R and sim_S into a single similarity matrix using equation 4

find top- n similar users U_1, \dots, U_n to U

get the corresponding similarity values of top- n similar users

compute the predicted ratings of user U for each unrated item using equation 3

find the top- e items (items with highest predicted rating) for user U and put them in O

if there is a product-search history corresponding to user U
then

 retrieve keywords from search history

 using Smith Waterman String Similarity coefficient calculate the similarity between the retrieved keywords and the keywords of each top- e items individually.

 reorder the top- e items in the decreasing order of their keyword similarity measure

 put top- e items in O

end if

if the predicted items are j less than the required items to be recommended then

add in O , j no. of those items which are recently added to the database (store)

end if

return O

7. EXPERIMENTAL EVALUATION

In this section, first we compare experimentally various string similarity methods and decide which similarity method would be useful in increasing the accuracy of our social hybrid recommender system. Then the relationship between top- e items (to be predicted) and the average item prediction accuracy of our recommender system, relationship between top- e items and average keyword-hit accuracy is shown. Lastly the relationship between Smith Waterman string similarity coefficient threshold and average keyword-hit accuracy is shown. All our experiments were performed on a 2.3 GHz Pentium Core 2 Duo processor with 4 GB of memory. All algorithms are implemented in Java programming language. To evaluate the results and to show the relationship between various factors, we have used artificial datasets (synthesized from automated programs).

For matching keywords various string similarity methods like Cosine similarity, Dice similarity [21], Euclidean distance, Jaccard similarity, Jaro Winkler [22] similarity coefficient, QGramDistance similarity coefficient, Smith Waterman [23] similarity coefficient can be used. To find which similarity method will give the best result, we performed an experiment using some sample strings/sentences. String 1 is “web database applications with php and mysql third edition TMH”. String 1 contains keyword regarding a book. String 2 is “web database applications”. Keywords for string 2 can be obtained from product-search history from user. String 3 is “php database applications tmh” and string 4 is “web applications”.

The results of our experiment can be easily seen through chart 1, that Smith Waterman string similarity method provides the best similarity value in each of the three cases i.e., between string 1 & string 2, string 1 & string 3, string 1 & string 4. It can be also verified that whether we pass the keywords in the form of string to calculate the string similarity or we calculate the similarity of individual keywords and then take the average similarity value of all the keywords, Smith Waterman string similarity method still outperforms all other methods.

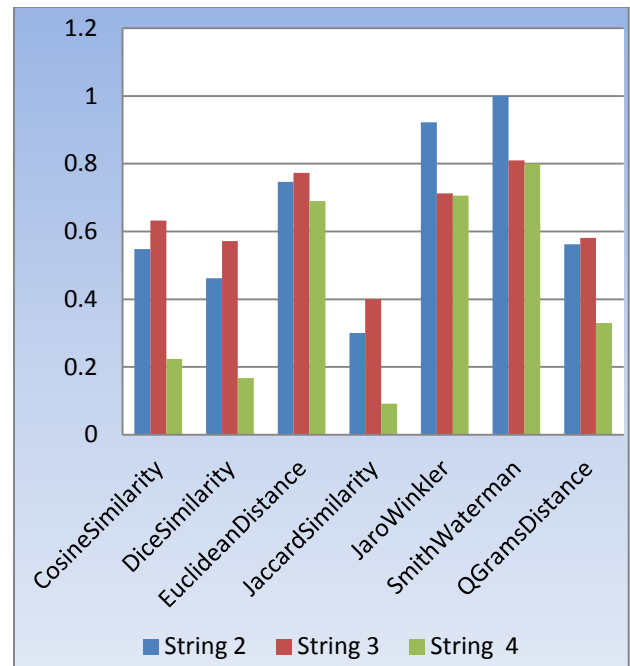


Chart 1: shows the similarity of string 1 with other strings

To evaluate the efficiency of our recommender system two facts can be considered. First one is “how many correct recommendations our recommender system can generate” and second is “how quickly our recommender system can generate recommendations”. Most of the studies on recommender systems consider recommender quality over computation competence because computation competence bottleneck can be removed by using faster processors, high performance hardware and additional memory. To test the recommendation quality of our recommender system, we first let the target user to mark the top- e unrated items and then compared these marked items with our system generated top- e items.

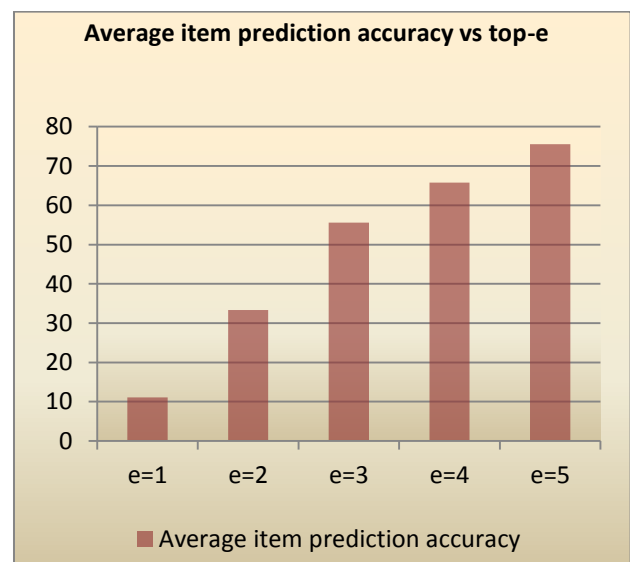


Chart 2: shows average item prediction accuracy vs top-e

Chart 2 shows the increase in average item prediction accuracy on increasing the top- e items (items to be recommended). The system used keyword hit criteria to reorder items to be recommended. A similar chart 3 is drawn

which shows the increase in keyword-hit accuracy as the top-e value increases. Since our proposed system have used Smith Waterman string similarity method to find similarity among keywords, therefore some threshold criteria should be set such that keywords whose similarity value is below that threshold should not be considered to calculate the average keyword-hit accuracy. Chart 4 shows the decrease in average keyword-hit accuracy as the threshold value increases. The threshold value (I) in chart 4 is shown in percentage.

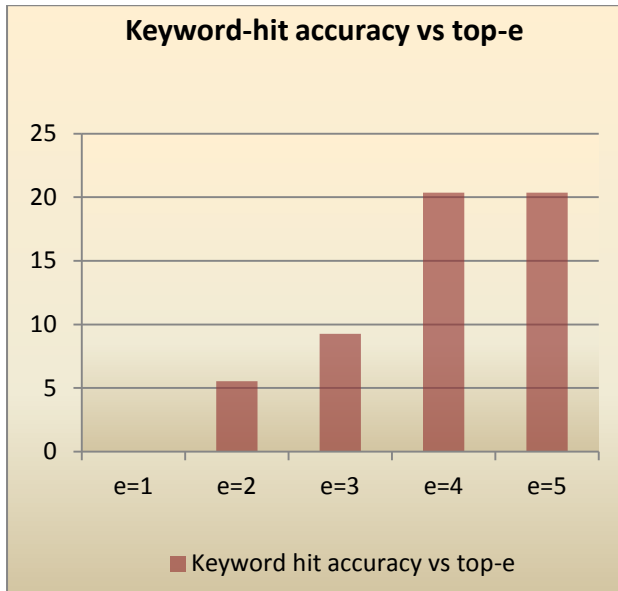


Chart 3: shows keyword hit accuracy vs top-e

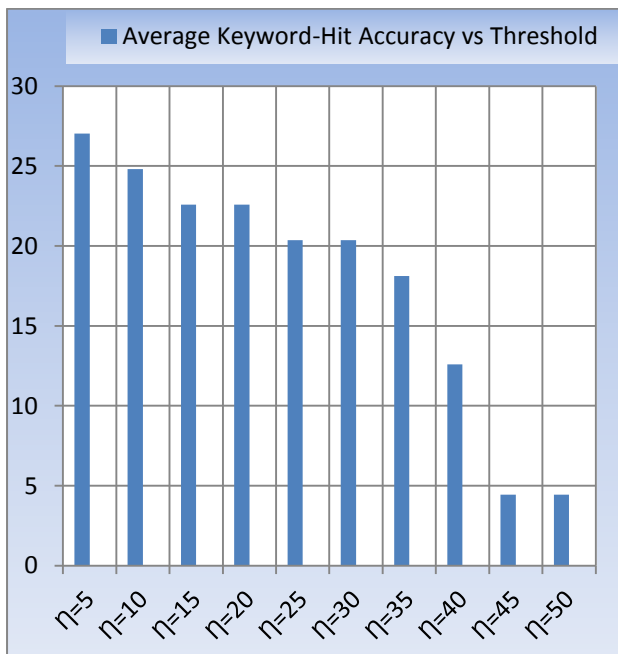


Chart 4: shows average keyword-hit accuracy vs threshold

8. CONCLUSION AND FUTURE SCOPE

In this paper, we presented a new social hybrid algorithm to make product recommendations in the online environment. Our social hybrid product recommender algorithm unifies the similarity matrices obtained from both user-item rating network and friend's network. The system calibrates the

influence of each social network for each target user. We performed broad experimental evaluation of our recommender system. We have experimentally shown how average item prediction accuracy and keyword-hit accuracy varies with top-e items to be recommended. We have shown that Smith Waterman string similarity method outperformed all other methods mentioned in the previous section. Our social hybrid product recommender can deal with data sparsity problems and works even when explicit trust rating data is not available.

Future work could include factors such as "like/dislike", "share", "tag" and "comment" to calculate the user-user similarity matrix, testing the effectiveness of our proposed recommender system against a large real life social network datasets to verify its suitability in social commerce environment.

9. REFERENCES

- [1] Konstan, J. A., B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, vol. 40, no. 3, pp. 77-87, 1997.
- [2] Resnick, P., N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM conference on Computer Supported Cooperative Work*, pp. 175- 186, New York, NY, USA, 1994.
- [3] Sarwar B., G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of WWW Conference*, pp. 285-295, 2001.
- [4] Karypis G. Evaluation of item-based top-n recommendation algorithms. In *Proc. ACM CIKM Conf.*, pp. 247-254, 2001.
- [5] Billsus, D. & M. Pazzani. A Hybrid User Model for News Story Classification. In *Proceedings of 7th International Conference on User Modelling*, pp. 99-108, Banff, Canada, 1999.
- [6] Billsus, D. and M. Pazzani. User modeling for adaptive news access. *User Modeling and User-Adapted Interaction*, vol. 10, no. 2-3, pp. 147-180, Kluwer Academic Publishers Hingham, MA, USA, 2000.
- [7] Tran, T. and R. Cohen. Hybrid Recommender Systems for Electronic Commerce. In *Knowledge-Based Electronic Markets, Papers from AAAI Workshop*, Technical Report WS-00-04, pp. 78-83, AAAI Press, 2000.
- [8] Pazzani, M. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, vol. 13. No. 5-6, pp. 393-408, December 1999.
- [9] Balabanovic, M. and Y. Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, vol. 40, no. 3, pp. 66-72, 1997.
- [10] Basu, C., H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI'98)*, pp. 714- 720, Madison, Wis, USA, July 1998.
- [11] Popescul, A., L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic Models for Unified

- Collaborative and Content-Based Recommendation in Sparse-Data Environments. In Proceedings of the 17th Conf. on Uncertainty in Artificial Intelligence (UAI'01), pp. 437- 444, Seattle, WA, 2001.
- [12] Schein, A. I., A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and Metrics for cold-start recommendations. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02), pp. 253-260, ACM press, New York, NY, USA, 2002.
- [13] Melville, P., R. J. Mooney, and R. Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations. In Proceedings of the Eighteenth National Conference on Artificial Intelligence, Edmonton, pp. 187-192, AAAI press, Menlo Park, CA, USA, 2002.
- [14] Soboroff, I. and C. Nicholas. Combining content and collaboration in text filtering. In 43 IJCAI'99 Workshop: Machine Learning for Information Filtering, Stockholm, Sweden, August 1999.
- [15] J. Golbeck. Personalizing applications through integration of inferred trust values in semantic web-based social networks. In Proceedings of Semantic Network Analysis Workshop at the 4th International Semantic Web Conf., 2005.
- [16] Massa, P. & P. Avesani. Trust-aware collaborative filtering for recommender systems. In Proceedings of Federated International Conference on The Move to Meaningful Internet: CoopIS, DOA, ODBASE, pages 492–508, 2004.
- [17] Jamali, M. & M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of 4th ACM RecSys Conference, pages 135–142, 2010.
- [18] He, J. & W. Chu. A Social Network-Based Recommender System (SNRS). In Annals of Information Systems, Special Issue on Data Mining for Social Network Data, Vol.12, pages 44–74, 2010.
- [19] McPherson, M., M. Smith-Lovin & J. M Cook. Birds of a Feather: Homophily in Social Networks. In Annual Review of Sociology, vol 27, pp. 415-444, 2001.
- [20] Symeonidis, P., E. Tiakas and Y. Manolopoulos. Product Recommendation & Rating Prediction based on Multi-modal Social Networks. In Proceedings of the 5th ACM conference on Recommender systems, pp. 61-68, New York, NY, USA, 2011.
- [21] L. Dice. Measures of the amount of ecologic association between species. Ecology, vol. 26, no. 3, 1945.
- [22] W. E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In Proceedings of the Section on Survey Research Methods, American Statistical Association, pp. 354–359, 1990.
- [23] Smith, F. T. & S. M. Waterman. Identification of Common Molecular Subsequences. Journal of Molecular Biology, vol. 147, pp. 195–197, 1981.