

Increasing Recommendation Accuracy and Diversity via Social Networks Hyperbolic Embedding

Vasiliki Pouli

University of Maryland
Institute for Systems Research,
College Park, MD 20742, USA
vpouli@umd.edu

John S. Baras

University of Maryland
Institute for Systems Research,
College Park, MD 20742, USA
baras@umd.edu

Anastasios Arvanitis

National Technical University of Athens,
Iroon Polytechniou, Zografou 157 80
Athens, Greece
anarv@dblbn.ntua.gr

Abstract—Several applications are built around sharing information by leveraging social network connections. For example, in social buying sites like Groupon, a deal is usually forwarded to interested recipients through their social graph. A primary goal is to improve user satisfaction by maximizing the relevance of the shared message to the target audience. In order to suggest more personalized products, one should consider offering not only accurate but also diverse recommendations, since diversification plays an important factor in increasing the users's satisfaction. In this work, we address this problem by proposing a social network hyperbolic embedding that exploits both social connections and user preferences aiming at increasing both the accuracy and the diversity of recommendations.

I. INTRODUCTION

The increase of online commercial environments offers users millions of choices, which makes it hard and time consuming to evaluate them in order to select a service or product. Moreover, promoting products is often considered as spam, especially if the recommended products are irrelevant to the recipient. Thus, a primary challenge is how to effectively recommend a product only to users that are the most likely to find this product interesting, with the goal to increase both user satisfaction and profit [20].

For this purpose, several modern applications could leverage social network connections in order to promote products. For example, social buying sites like Groupon.com or auction websites such as eBay.com could use a very effective way to promote their products or services by sharing product recommendations to users and their friends via social networks such as Facebook [8], Twitter [3] etc.

Recommender systems are used to support the user decision-making process by providing suggestions to users about what to buy, which movie to watch, what

to listen, where to travel, etc. For this reason they have become a valuable tool for online platforms to help customers in their planning and purchasing decisions. For example in eBay or Amazon the platform recommends the most relevant items to each user based on her search queries and the behavior of like-minded users. Another example is movie rental applications such as Netflix, where the platform suggests movies according to users' previous ratings and other users' similar tastes.

Which factors have to be considered when recommending things? And which are the most successful techniques to generate recommendations to interested users, such that we avoid spamming everyone with irrelevant information? The problem of providing accurate recommendations to users is generally complex; Netflix announced the Netflix Prize competition (netflix-prize.com) [4] with a prize of \$1M for those who would improve the accuracy of predictions by 10%.

Recommendations can be done via various methods, however among the most well-known methods is Collaborative Filtering (CF) [6]. CF is a method that can be used to select items which are similar to other items that a user has rated (item-based CF), as it is used in Amazon.com [14], [16] or by utilizing information collected from users with similar interests (user-based CF) [21], as seen in GroupLens. The first step of collaborative filtering measures the similarity of users/items based on various similarity measures such as cosine similarity or Pearson correlation that we will discuss in Section 2. The k users/items that have the highest similarity with the active user/item are selected; these are usually referred to as neighbors. Finally, the ratings of the neighbors are aggregated and used to predict the user's rating.

In order to improve the accuracy of recommendations, one has to consider how relevant a product or service

is to a user. This relevance score is often context-sensitive; i.e., it depends on several factors represented as dimensions. Such dimensions could be for example the thematic category, price, location of the product. If we consider, e.g. a coupon for a Chinese restaurant in Washington DC, it would be wise to forward it mainly to users that are located near Washington DC and are interested in Chinese cuisine. Thereby, it is more likely that these users will accept the recommended deal and eventually buy it. On the other hand, flooding a message concerning a product or service to social connections of a user, including those that live far away from Washington DC or have not expressed any interest for food coupons, would probably be considered as spam. The dependence of relevance to the message context [18] calls upon developing a dynamic routing approach.

Moreover, when recommending products or items an important factor that should be also taken into account apart from accuracy is diversity [15], [22]. The importance in having diverse recommendations has been highlighted in several studies (e.g. [1], [2], [5]). Increasing the diversity in recommendations can help companies sell a variety of products in addition to the highly ranked ones. It could also be beneficial for some providers that want to promote long tail items, i.e. items that are not so popular in the sales distribution either because they are older or because they are low budget ones.

In this work we propose building a hyperbolic embedding of a user friendship network by bringing similar users closer together. This will ensure message delivery to all the recipient nodes of the social network and produce better accuracy and diversity compared to the traditional CF techniques. The model considers user preferences and delivers product recommendations to the nodes that are the most interested, thus offering better personalization to the users.

II. BACKGROUND INFORMATION AND RELATED WORK

A. Social-based Routing Algorithms

Recent works [10], [11], [12] have proposed routing algorithms that use social characteristics with applications on delay tolerant, ad hoc or P2P networks. In [12] the authors propose a socially-based greedy routing algorithm for delay tolerant networks. They use an n -dimensional social profile of users and utilize the Jaccard coefficient to measure the similarity between them. Then, they apply a greedy routing algorithm by selecting the nodes that are socially closer to each other. Also, in [11] the authors proposed the Bubble rap, where they utilized nodes' centralities and communities to improve the

forwarding efficiency. Furthermore, in [10], the authors have shown that labeling the nodes with their affiliation and forwarding messages to nodes belonging to the same affiliation as the destination, can improve forwarding performance both in terms of delivery ratio and cost. However, although these algorithms exploit user-to-user similarities, they (i) do not utilize their preferences in terms of the forwarded message content, they (ii) do not ensure message delivery to all interested recipients and they (iii) do not take into account social friendship connections. Motivated by this, in our current work we propose a context-based routing approach that exploits both user preferences along with the social network friendship connections and creates a network structure aiming at maximizing relevance and potential profit by ensuring greater accuracy and diversity in recommendations to the most interested nodes.

B. Recommendation Systems.

Recommendation systems have proved very helpful for users that wish to find interesting items from a very large information and product space. One of the most common methods used is collaborative filtering (CF) where similar users or items are identified and then the top- k most highly ranked among them are selected to be used for final predictions. The basic idea for collaborative filtering lies in calculating the similarity between each user/item by comparing the common ratings of each user/item with other users/items. The most common similarity measures used in collaborative filtering are:

- *Cosine similarity* where user/item ratings are represented as points in a vector space model and $\cos\theta$ is calculated between them. Depending on whether the similarity is measured on items or users, the cosine similarity can be either item-based or user-based. Item-based cosine similarity is computed by considering co-rated items only. These co-rated pairs are obtained from different users. Item-based cosine similarity is defined as follows:

$$\text{sim}(i, i') = \frac{\sum_{u \in U(i, i')} R(u, i) * R(u, i')}{\sqrt{\sum_{u \in U(i, i')} R(u, i)^2} \sqrt{\sum_{u \in U(i, i')} R(u, i')^2}} \quad (1)$$

where $\text{sim}(i, i')$ is the similarity of item i with item i' , $R(u, i)$ is the rating of user u for item i , $R(u, i')$ is the rating of user u for item i' and $U(i, i')$ is the set of all users that have rated both items i and i' as seen in Figure 1.

User-based cosine similarity can be computed in a similar way by interchanging items with users.

•**Pearson correlation coefficient:** The similarity between items can be also computed based on the Pearson correlation which measures the linear relationship between objects and is calculated using the following equation:

$$sim(i, i') = \frac{\sum_{u \in U(i, i')} (R(u, i) - \bar{R}(i)) * (R(u, i') - \bar{R}(i'))}{\sqrt{\sum_{u \in U(i, i')} (R(u, i) - \bar{R}(i))^2} \sqrt{\sum_{u \in U(i, i')} (R(u, i') - \bar{R}(i'))^2}} \quad (2)$$

where $sim(i, i')$ is the similarity of item i with item i' , $R(u, i)$ is the rating of user u for item i , $R(u, i')$ is the rating of user u for item i' , $\bar{R}(i)$ is the mean rating of item i , $\bar{R}(i')$ is the mean rating of item i' and $U(i, i')$ is the set of all users that have rated both items i and i' as seen in Figure 1. The Pearson Correlation measures the strength of the linear relationship between two variables and takes values in the range $[-1, 1]$, where -1 shows a negative correlation, 0 means no correlation, and 1 is a positive correlation.

Several variations of the above similarity measures have been also proposed in the literature including adjusted weighted schemes, Spearman rank correlation, Kendall's Correlation, mean squared differences, entropy [19]. Spertus et al. [17] conducted a large-scale study to evaluate six different similarity measures on the Orkut social network and their experiments showed that the best results in recommendations were those produced based on the cosine similarity.

Although accuracy is usually the target of the recommendation systems [4], the produced recommendations are not always useful for the users who often wish for something different or diverse. Thus, diversity should also be taken into account [1], [5], [15]. With this goal, Bradley and Smyth [5] have proposed a method to improve recommendation diversity by considering it as the average dissimilarity between all pairs of items in the result-set. We will use the diversity metric as proposed in [1], [2]. In these works Adomavicius and Kwon adopt CF ranking-based heuristics to improve the aggregate recommendation diversity without affecting the accuracy. They achieve this by recommending items whose ratings are above a specific threshold. In our work, we consider the same diversity measure but examine a different accuracy metric that does not take into account the rating thresholds, since it sometimes limits the available options. We create a hyperbolic network and analyse its recommendation performance by comparing it with traditional CF techniques. We show that our model produces better accuracy and diversity measures compared to the traditional CF methods, when forwarding relevant

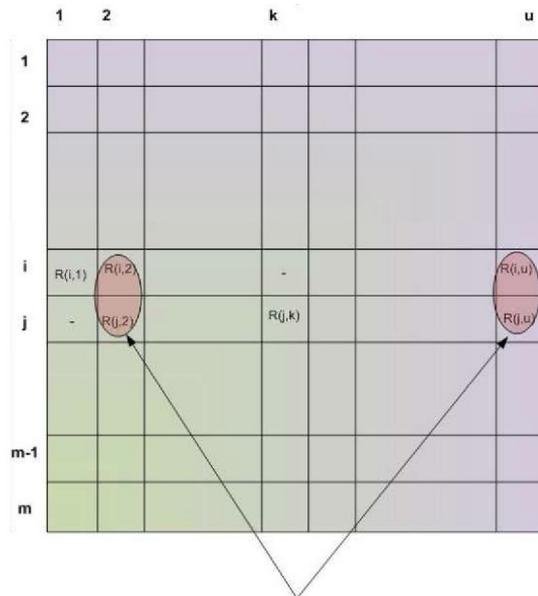


Fig. 1. User (columns) - Item (rows) rating matrix

product recommendations to the most interested social connections.

III. RECOMMENDATIONS IN HYPERBOLICALLY EMBEDDED SOCIAL NETWORKS

In this section we will describe our similarity measure that is used for calculating node-message relevance distances. Our similarity measure consists of two parts; the first part incorporates the context similarity, whereas the second part measures the (structural) network distance. Subsequently, we will present our context-based routing algorithm that utilizes our proposed node similarity measure in order to propagate the message to all interested recipients.

A. User-item context model

We will first explain the context part by describing the context model that we used. We will assume that both the product message and a users' interests/preferences can be represented using the same context model. This context information can be modeled as a vector of attributes where each attribute represents a context dimension. In general, the domain for each context dimension might take values from a hierarchy, such as the one depicted in Figure 2. Such context dimensions could be for example the content topic (e.g. food, entertainment, traveling, movie genre), a location, price range, etc.

We assume that each user has specific interests which can be defined using a set of keywords. These keywords

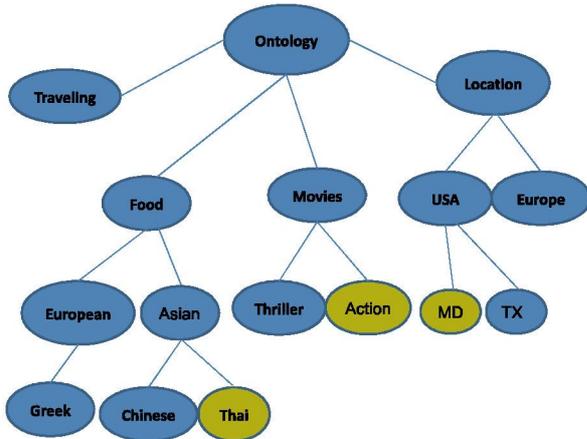


Fig. 2. A Context Hierarchy

might take any value from the attribute domain. For example a user, as seen in Figure 2, might have specified that she likes 'Thai' cuisine and 'Action' movies in 'Baltimore, MD'. These preferences can be explicitly provided by each user (such as Facebook 'likes') or they might as well have been extracted by applying data mining techniques on the user's posts, comments, shares etc. We also consider a product item that has values taken from the same context model with a product vector $I_j = \{Asian, Baltimore\}$, which indicates that it refers to a deal about Asian food in a restaurant located in Baltimore, MD. We could also define weights in the context dimensions and say for example that a user likes Asian cuisine with a weight score of 80%. As user preferences we could use the items (e.g. books, movies, music) that users recommend to their friends in a social network such as the Douban.com chinese network. In such a network a user can become a friend or fan of another user, collect interesting movies, books and music albums, rate them or recommend them to his friends and fans. The friends of a user can see a social recommendation and either accept it by collecting the item later, or choose to decline it.

In order to measure the context similarity between a message and a user's preferences, we will use a cosine similarity between the preference vector of user U_i and the product vector I_j .

$$context_similarity(U_i, I_j) = \frac{U_i \cdot I_j}{\|U_i\| \|I_j\|} \quad (3)$$

B. User-user follower Similarity Measure

In this section we introduce a new similarity measure to estimate the similarity between nodes in a social

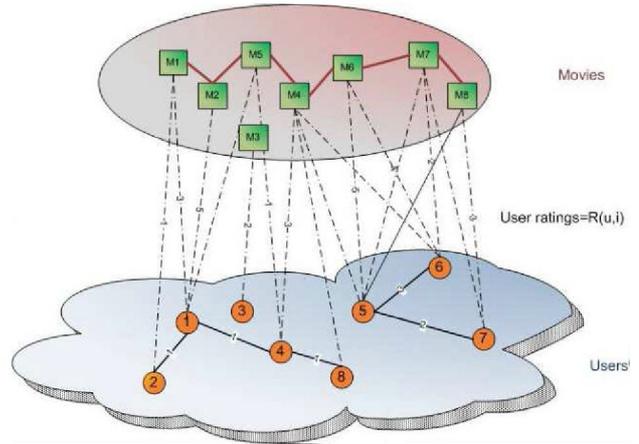


Fig. 3. User similarity network

network. We will try to identify who will be the most similar among them according to their preferences. We call our new similarity measure as *follower's similarity measure* and we define two versions: one that takes into account the user ratings (*rating-based*) and the other one that does not take ratings into account (*non rating-based*).

1) *Non Rating-based*: Usually in a social network nodes are connected with friendship connections or with connections that specify who follows whom where nodes follow each other based on their common interests or ratings. We adapt these 'follower' connections by introducing some weights according to the following rule: two connected users have k edge weight if they have rated k common items. k will be the edge weight and similarity measure between them, normalized when divided by the total number of items thus:

$$sim(u, u') = \frac{|I(u, u')|}{m_t} = k/m_t \quad (4)$$

For example, in Figure 3 where we assume a movie recommendation system where users can rate the movies they have watched, users 1 and 2 are connected with weight 1 since both of them have rated one movie (M1).

2) *Rating-Based*: In the second version we take into account the user ratings for the items and compute the edge connection weight between the two users as disproportional to the difference of the ratings for the common items they have rated. The more the users differ in their ratings the more dissimilar they are.

$$sim(u, u') = \frac{1}{1 + \sum_{i \in I(u, u')} |R(u, i) - R(u', i)|} \quad (5)$$

where $I(u, u')$ is the set of items rated by both user u and u' .

For example user 1 is connected with user 4 since both of them have rated movie 5 (M5) (1 common movie). Note that user 1 has also rated movie 1 (M1) and movie 2 (M2) and user 4 has rated movie 4 (M4). These movies might not be similar; however the fact that both users 1 and 4 have rated M5 suggests that they might be interested in each other's different choices (M2 and M4) as well and follow them in order to increase the diversification of their recommendations.

C. Recommendation Predictions

By creating user-user follower networks and the edge weights between each two nodes, we infer their similarity and use it as a similarity measure instead of Equation 1 to predict ratings as described below in Equation 6. Using our model a network with weighted edges is created and users can follow their most similar neighbors' top recommendations to increase the diversity of their recommendations. In order to estimate the rating $R^*(u, i)$ that user u would assign to an item i we compute $R^*(u, i)$, which is the adjusted weighted sum of all ratings $R(u', i)$, where u' belongs to $K(u)$: the K neighbors of user u who rated item i and have the highest similarity $sim(u, u')$ to user u . $\bar{R}(u)$ is the average rating of user u and $\bar{R}(u')$ of user u' respectively. This weighted sum can capture how the users rate common items and is defined as follows [2]:

$$R^*(u, i) = \bar{R}(u) + \frac{\sum_{u' \in K(u)} sim(u, u')(R(u', i) - \bar{R}(u'))}{\sum_{u' \in K(u)} |sim(u, u')|} \quad (6)$$

D. Accuracy vs. Diversity

As discussed above, by increasing the diversity in recommendations we could have more personalized items and companies could also increase their revenues by offering long tail products. Users could also have diversity in taste by following their friends actions. We would like to measure the accuracy and diversity variations under different scenarios.

By *accuracy*, as in [2], we measure the statistical precision measure, which we consider as the percentage of items from the *top-k* high-ranked predictions measured from (6) which are indeed rated as the *top-k* in the original ratings of each user.

$$accuracy@k = |correct(S_k(u))|/k \quad (7)$$

where $S_k(u)$ is the set of the k highest ranked predicted items for user u .

By *diversity* we define the union of the different *top-k* highest rated items predicted from eq. (6) compared to

the items that were rated as *top-k* from the original user ratings, across all users:

$$diversity@k = |S_k(u) - S'_k(u)| \quad (8)$$

where $S_k(u)$ is the set of the k highest ranked predicted items for user u and $S'_k(u)$ is the set of the k items that were originally ranked as *top-k* in the original user ratings of user u .

E. Hyperbolic Embedding

Furthermore, in order to ensure message delivery to all interested recipients, our approach greedily embeds the network into the hyperbolic space. We followed the greedy hyperbolic embedding since, as it is shown in [13], every finite, connected, undirected graph has a greedy embedding in a two-dimensional hyperbolic space, i.e., *one may achieve 100% success rate with greedy routing by assigning virtual coordinates in the hyperbolic plane rather than the Euclidean plane*. Based on this embedding, each node will always have a neighbor closer to the destination and a message will never get stuck into local minima as in Euclidean space. In order to embed the network into the hyperbolic space we applied the algorithm described in [7]. Following this approach, we constructed a spanning tree from the original graph, and assigned to each node hyperbolic coordinates from the set $\mathbb{D} = \{z \in \mathbb{C} \mid |z| < 1\}$ in accordance to their parents coordinates. A greedy embedding of the spanning tree is also a greedy embedding of the graph. We consider as network distance $network_dist(z_1, z_2)$ between two nodes z_1 and z_2 the hyperbolic distance between them. Figure 4 shows a random graph of 20 nodes and Figure 5 its minimum spanning tree hyperbolic embedding in the Poincaré half plane disc.

F. Routing Algorithm

The goal of the routing algorithm is to ensure that each message will reach, in the minimum number of steps, the nodes that are most relevant to a message. We will assume that a product vendor wants to forward the message through his friends/followers' connections to the *top-m* most interested neighbors in the minimum number of steps. Based on the proposed context similarity measure and hyperbolic embedding mapping, we defined a relevance metric, named: *relevance*, given by Equation 9. This metric incorporates both the context similarity and the (structural) network distance between user i and u' and calculates the relevance of a node i to an item j :

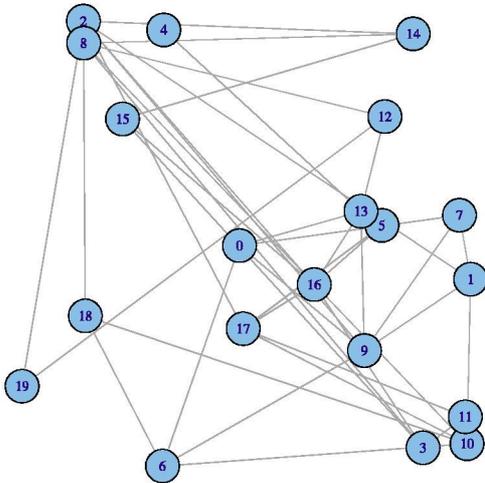


Fig. 4. Random graph with 20 nodes

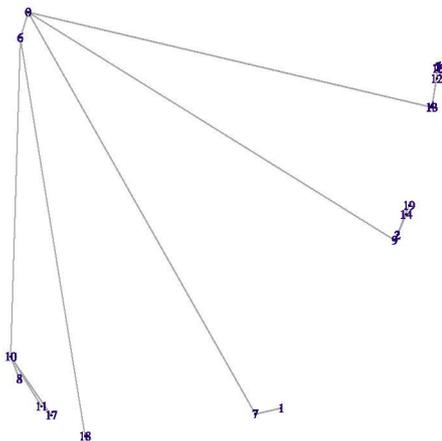


Fig. 5. Greedy Hyperbolic embedding of the minimum spanning tree of the graph depicted in Figure 4 in the half plane Poincaré disc

$$\begin{aligned} \operatorname{argmax}_i \operatorname{relevance}(i, j) \\ = \frac{\operatorname{context_similarity}(U_i, I_j)}{\operatorname{network_dist}(i, u')} \quad (9) \end{aligned}$$

In order to discover these nodes, we can simply evaluate Equation 9 on the nodes of the network and identify the nodes that are hyperbolically closer and most similar to the source node with regard to the user-item context similarity measure along with the user-user similarity measure and the network distance available through the hyperbolic graph embedding. The target

nodes i.e. the users where the message will be forwarded to, are the ones that maximize our similarity metric: $\operatorname{relevance}(i, j)$.

After identifying the target nodes, we proceed to our routing algorithm. Initially, the algorithm begins from D , i.e., the node that issues the message. For example in Facebook [9], D might be the account of a service provider. Next, at each step the algorithm utilizes our network embedding to find out which of the neighbors are closer to the user than the others. From these nodes, the algorithm selects to forward the recommendation to the n neighbors that have the highest relevance score based on Equation 9.

IV. EXPERIMENTAL EVALUATION

We conducted a set of experiments to examine the efficiency of our algorithm which chooses nodes most relevant to the message against a non social-based scheme, where nodes choose neighbors to forward the message to without taking into account their preference similarity. We used both synthetic and real data to test the efficiency of our model.

A. Synthetic Network

For the synthetic network we assume that a product provider who wants to recommend a product could specify under which categories and sub-categories the product belongs to, e.g., Asian food, thriller movies, traveling to Hawaii etc. The subscribers could as well have a similar menu that would allow them to select for which categories they would like to receive messages/products. Thereby, the system will receive each item and forward it to the recipients that have shown interest to the particular categories associated with the product.

For the creation of the synthetic network we used the R Project (www.r-project.org) and generated an undirected random network with $N=100$ nodes and probability for adding an edge between two arbitrary vertices equal to 0.02. We chose $m=1$ meaning that we take one destination i.e. the node i that maximizes $\operatorname{relevance}(i, j)$. Further, we also used a fanout = 1 i.e., our routing algorithm will pass a recommendation to the one neighbor that maximizes his context similarity with the product and has a small network distance from the destination node.

In each experiment we selected random user preference and product values for the context dimension categories and we executed the simulation 100 times by selecting a different source node for every iteration.

Through our simulations we want to compute the path preference $\operatorname{pathPref}$ which we define as the average

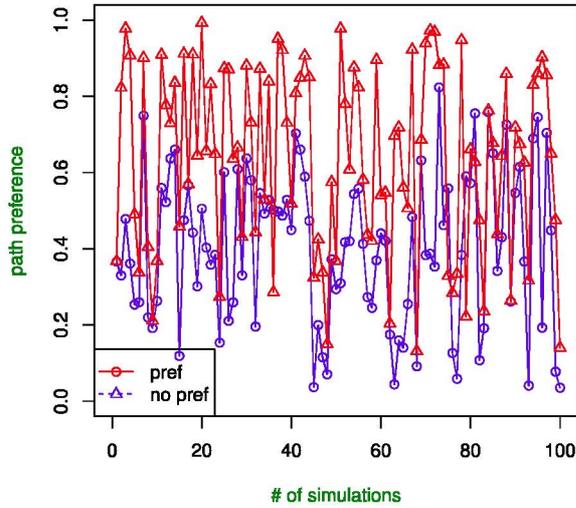


Fig. 6. Path preference

context similarity of the coupon with all the nodes along the path:

$$pathPref = \sum_{i=1}^M \frac{context_sim(i,j)}{M} \quad (10)$$

where M is the number of hops along the path.

The motivation for using path preference as a metric in our experiments, is that we want to evaluate the relevance of the message with the users that belong to the path followed. The closer a user is to a producer node in the social graph, the more likely it is to like the products he offers, which increases the probability of buying those products. Thus, path preference metric could be considered as a profit measure.

We calculate the path preference from the source to the destination using our proposed context-based routing algorithm, and then we compare it against a greedy forwarding routing which selects to forward a message recommendation to the neighbor closest to the destination but without taking into account the preferences of the nodes.

According to Figure 6 we observe that by exploiting the user preferences in our algorithm, then the whole path preference increases by 94%, compared to the simple greedy case, which does not take into account the similarity of the user with the product.

B. Evaluation of Recommendation Diversity and Accuracy in Real Network Data

To test the effectiveness of our model in a real network we executed experiments using a real dataset provided by the Douban.com social site. The dataset consists of 2000 items rated by 1000 users. From eq. (6)

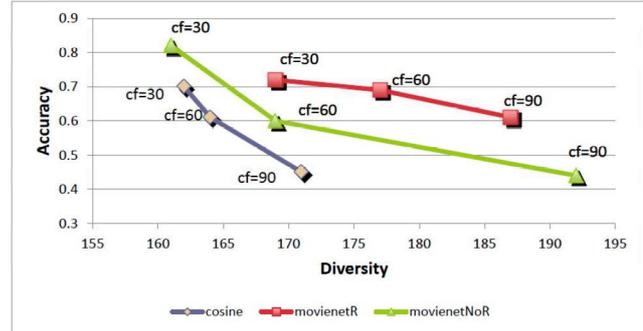


Fig. 7. Diversity vs Accuracy

we compute the top-3 highly predicted items for each user. The algorithm then tests how the mean accuracy and the overall diversity varies for the top-3 highly predicted items forwarded to the most relevant node, that is the most relevant node for a specific source user and forwarded item. We measure the average accuracy along all users in the path as the mean of the accuracies defined in eq. (7) across all users and the overall diversity as the union of the diversities defined in (8) for all users. The algorithm starts from random source nodes each time and follows different routes depending on the relevance of each item to the users. To find the most relevant node for each item we used three different similarity metrics, namely the user-based cosine similarity - eq. (1) (seen as 'cosine' in Figure 7) and the other two by considering the proposed non rating-based ('followerNoR') and rating-based ('followerR') user-follower similarity network - eq. (4) and (5). In each scenario, as seen in Figure 7, we used an increasing number of neighbors ranging from 30 to 90 with a step of 30 each time. From the measurements we can see the tradeoff between accuracy and diversity in these three scenarios: the accuracy is high when diversity is low and then decreases as the diversity increases; meaning that the more accurate recommendations we have the less diverse options we offer to the users. This is normal since the more accurate recommendations are typically safer and closer to the user's expressed interests.

Further, we can observe that the two versions of our proposed user-follower network exhibit better diversity than the conventional cosine similarity. We chose to test our model for various numbers of neighbors to show how it affects the tradeoff between the accuracy and diversity. We found out that the more neighbors selected the less accuracy is achieved but with greater diversity, since the sources of recommendations are wider by selecting a greater amount of people that provide options not so

similar to a user. From the measurements we can see that compared to the conventional cosine similarity, the two versions of the user follower network have better diversity and accuracy for the same number of neighbors used.

V. CONCLUSION

In this paper we proposed a context-aware routing scheme that aims to increase the relevance of messages shared across a social network. This is achieved by forwarding the message to the most relevant nodes, taking into account both user preferences and similarity along with the network structure. Our experiments show that our context-aware routing scheme provides significant improvements in terms of the preference similarity of nodes along the path that is followed, thus increasing the chances of users that might be interested to the message context to buy a product. In order to identify the similar users we described a social connection network hyperbolic embedding that aims at creating weighted connections between users to increase the overall diversity and accuracy of the items recommended to most interested and similar nodes. Our model is flexible since it can be adopted by the widely used CF technique and provides a follower's connection measure for selecting the right neighbors according to the system needs. If we aim at improving the accuracy we should create a rating-based connection between users. On the other hand, if greater diversity is required then the non-rating based approach should be followed. We adopted the steps of the CF technique to make predictions and saw that, compared to the traditional cosine similarity used to select neighbors, our proposed approach yields better results by offering greater accuracy and diversity.

ACKNOWLEDGEMENT

Research partially supported by the US Air Force Office of Scientific Research MURI grants FA9550-10-1-0573 and FA9550-09-1-0538.

REFERENCES

- [1] G. Adomavicius and Y. Kwon. Overcoming Accuracy-Diversity Tradeoff in Recommender Systems: A Variance-Based Approach. In *Proc of the 18th Workshop on Information Technology and Systems (WITS'08)*, 2008.
- [2] G. Adomavicius and Y. Kwon. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE TKDE*, 24(5):896–911, 2012.
- [3] I. Anger and C. Kittl. Measuring influence on twitter. In *I-KNOW*, page 31, 2011.
- [4] J. Bennett and S. Lanning. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, 2007.
- [5] K. Bradley and B. Smyth. Improving recommendation diversity. In *Proc. 12th Irish Conf. Artificial Intelligence and Cognitive Science*, 2001.
- [6] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52, 1998.
- [7] A. Cvetkovski and M. Crovella. Hyperbolic embedding and routing for dynamic graphs. In *INFOCOM*, pages 1647–1655, 2009.
- [8] N. B. Ellison, C. Steinfield, and C. Lampe. The benefits of Facebook "friends: " social capital and college students' use of online social network sites. *J. Computer-Mediated Communication*, 12(4):1143–1168, 2007.
- [9] J. Feiler. *How to do everything : Facebook applications*. McGraw-Hill, New York, 2008.
- [10] P. Hui and J. Crowcroft. How small labels create big improvements. In *PerCom Workshops*, pages 65–70, 2007.
- [11] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions in Mobile Computing*, 10(11):1576–1589, 2011.
- [12] K. Jahanbakhsh, G. Shoja, and V. King. Social-greedy: a socially-based greedy routing algorithm for delay tolerant networks. In *MobiOpp*, pages 159–162, 2010.
- [13] R. Kleinberg. Geographic routing using hyperbolic space. In *INFOCOM*, pages 1902–1909, 2007.
- [14] G. Linden, B. Smith, and J. York. Industry report: Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Distributed Systems Online*, 4(1), 2003.
- [15] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI Extended Abstracts*, pages 1097–1101, 2006.
- [16] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.
- [17] E. Spertus, M. Sahami, and O. Buyukkokten. Evaluating similarity measures: a large-scale study in the Orkut social network. In *KDD*, pages 678–684, 2005.
- [18] S. Spiekermann, M. Rothensee, and M. Klaffit. Street marketing: how proximity and context drive coupon redemption. *Journal of Consumer Marketing*, 28(4):280–289, 2011.
- [19] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. Artificial Intelligence*, 2009, 2009.
- [20] P. K. Tat. A motivation-based model of coupon usage. *Journal of Promotion Management*, 3(1-2):1–14, 1996.
- [21] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *SIGIR*, pages 501–508, 2006.
- [22] M. Zhang and N. Hurley. Avoiding monotony: improving the diversity of recommendation lists. In *RecSys*, pages 123–130, 2008.