Mining Context Specific Inter-Personalised Trust for Recommendation Generation in Preference Networks

Quan Bai, Weihua Li and Jing Jiang

Auckland University of Technology, New Zealand
quan.bai@aut.ac.nz; weihua.li@aut.ac.nz; jing.jiang@aut.ac.nz

Abstract. This paper introduces a community-based approach to facilitate the generation of high-quality recommendations by leveraging the preferences of communities of similar users in preference networks. The proposed approach combines the idea of traditional recommendation systems and identification of network structures to explore context specific inter-personalised trust relationships among users. From the experimental results, we claim that the proposed approach can provide more accurate recommendations to individuals in a preference network.

Keywords: Community detection, preference network, recommender system

1 Introduction

In general, preference network represents the phenomenon that users have their own preferred items in a specific context, such as a social network; basically, it contains two types of elements, i.e., users and items [1]. Similar preference contributes to the trust relationships among the individuals, where trust refers to the level of belief established between two entities by considering past interactions in a certain context [2]. Whereas, the preference belief is normally subjective and conjuncted with certain context. Furthermore, trust can only be understood via observations and analysis as imperfect knowledge. Hence, it is difficult to explore certain objective behaviours of examined elements [2].

Recommender systems have emerged as an effective solution to the information overload problem [3]. In order to predict new likes or dislikes in preference networks, an automated recommendation approach is required for providing tailored and personalised information. However, traditional approaches, such as user-based collaborative filtering [4], item-based recommendation algorithms [5], only assume single and homogeneous trust relationships among the users, and evaluate item similarity from a simplistic world view. On the other side, community detection is widely applied to improve the accuracy for recommender systems, whereas, the available feedback ratings are ignored by many researchers [6][7]. Actually, this type of user-generated content is critical for perceiving users' preferences in a particular context.

In this paper, we propose a community-based recommendation approach for preference networks, which is capable of covering the aforementioned research gaps. The proposed algorithms explore context specific inter-personalised trust by modelling massive transactional data and analysing network structures. Specifically, the context specific inter-personalised trust indicates multiple and heterogeneous trust relationships among individuals in terms of different contextual situations. In other words, a particular user may place trust to different individuals in terms of their multi-faceted interests. The approach is motivated by the intuition that, according to the rating history of users, a group of users share the similar feedback records for same items, as they have similar preference and criteria for items. It leverages the features of traditional recommendation systems and network structures to explore context specific inter-personalised trust relationships.

The rest of this paper is organised as follows. In Section 2, trust estimation protocol and formal definitions are given. In Section 3, the hierarchical community structures are elaborated and community-based recommendation algorithms are presented. In Section 4, experimental results are given to demonstrate the performance of the proposed model by comparing with some traditional recommendation systems. Finally, the conclusion is presented in Section 5.

2 Community-based Trust Estimation Protocol

In this section, the trust estimation protocol is introduced, and the fundamental concepts are elaborated by giving formal definitions.

2.1 Trust Estimation Protocol

The protocol for community-based trust estimation approach is illustrated in Figure 1. There are six modules in the protocol, i.e., *the Reply Module, the Interaction Record Database, the User Criteria Clustering Module, the Facet Object Set Generation Module, the Prediction Retrieval Module and the Trust Calculation Module.* In this section, we will introduce the overall process in general.

Reply Module tends to collect the user-item ratings and store the **Interaction Records** IR into Interaction Record Database. The objective of User Criteria Clustering Module is to cluster users into hierarchical communities according to the user-generated ratings. Similarly, Facet Object Set Generation Module aims to create object communities based on the hierarchical user criteria clustering tree generated from the User Criteria Clustering Module. Prediction Retrieval Module handles the **Item Enquires** IE about a particular item $IE.item_j$ that the user does not have previous interactions with, by searching all the related facet object sets. Next, Facet Object Set Generation Module transfers the facet object sets to the Trust Calculation Module, whose objective is to produce a quality prediction for $IE.item_j$ based on the preference of enquiring user $IE.u_i$.

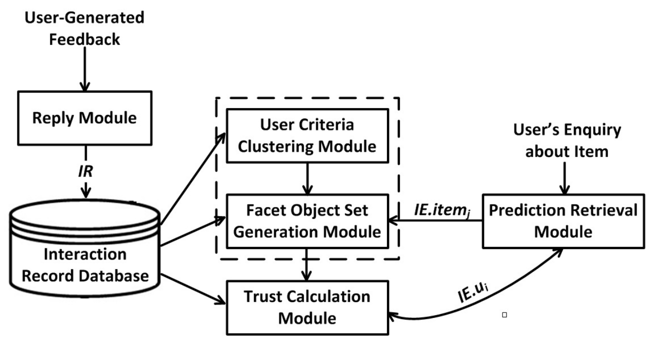

Fig. 1: Community-Based Trust Estimation Protocol

2.2 Formal Definition

In the current context, a preference network is comprised of an item set, i.e., $I = \{item_1, item_2, item_3, ..., item_n\}$, and a user set, i.e., $U = \{u_1, u_2, u_3, ..., u_m\}$. Each user may rate many different items, and every item can be rated by many users. Given a preference network having m users and n items, a $m \times n$ matrix R can be used to represent user-item ratings. Each entry $r_{m,n}$ in R denotes the feedback rating of item $item_n$ given by user u_m. $r_{m,n} = 0$, if u_m does not have any previous interactive experience with $item_n$.

Definition 1: Object Set O in a preference network is a set of objects. A particular object is represented as a two-tuple $o^{\tau}_{item_n} =< item_i, \tau_x >$, where $item_i \in I$, and τ_x denotes the rating value for $item_i$.

Once a pair of users, e.g., u_j and u_k, give a same rating τ_x to item $item_n$, the object $o^{\tau_x}_{item_n}$ is connected to both u_j and u_k. Thus, a preference network presents a bipartite pattern, consisting of two exclusive types of vertices representing users and the corresponding objects.

Definition 2: A Preference Network is a bipartite graph [8] represented as a three-tuple, i.e., $CG =< U, O, E >$, where U refers to the user set, O denotes the object set, and E indicates the edge set representing user-object interaction in CG, where $E = \{(u_j, o^{\tau_x}_{item_k}) | u_j \in U, o^{\tau_x}_{item_k} \in O\}$.

Definition 3: Interaction Record IR refers to the interactive feedback to a specific item $item_j$ given by u_i, which is represented as a three-tuple, $IR =< u_i, item_j, o^{\tau_x}_{item_j} >$.

If u_i inquires the potential quality of $item_j$, and u_i lacks of interaction experience with $item_j$, the system assembles an **Item Enquiry** $IE =< u_i, item_j >$, which indicates user $IE.u_i$ enquires item $IE.item_j$.

3 Hierarchical Community Structure

In this section, a four-step trust mining algorithm is proposed to partition different types of elements into community structures.

3.1 User Community

The user community detection is based on the intuitive fact that users in the same community more likely have similar expectations of a certain group of items. In this approach, items are regarded as random variables, and mutual information is capable of measuring general dependence among them. The entropy of a user rating pattern is a measurement of the uncertainty in feedback values given on items, which is formulated in Equation 1:

$$H(u_j) = -\sum_{i=1}^{n} P(R_{u_j} = r_{j,i}) \log P(R_{u_j} = r_{j,i}), \tag{1}$$

where n is the number of possible items which u_m rates. Higher entropy of users for item variables implies that their selection and rating pattern levels are more randomly distributed [9]. Mutual information describes the amount of common feedback ratings given by both users. Thus, the mutual information between user u_j and u_k is defined in Equation 2:

$$I(u_j, u_k) = H(u_j) + H(u_k) - H(u_j, u_k). \tag{2}$$

The smaller $I(u_j, u_k)$, the greater difference between pair of user selection and rating patterns. However, mutual information is not bounded, and it would not be a suitable distance measurement for itself. Therefore, we transform the mutual information into a bounded mutual-information-based distance by normalizing it (See Equation 3).

$$D(u_j, u_k) = 1 - \frac{I(u_j, u_k)}{\max(H(u_j), H(u_k))}. \tag{3}$$

In Equation 3, $D(u_j, u_k)$ denotes the preference similarity between a pair of users. $D(u_j, u_k) = 0$, if identical users have the maximum possible selection and rating patterns, as well as the identical entropies, i.e., $H(u_j) = H(u_k) = I(u_j, u_k)$ [10]. Hence, given a user set with m users, an $m \times m$ mutual-information-based distance matrix can be calculated by using Equation 3.

The user criteria clustering analysis algorithm is shown in Algorithm 1. In this algorithm, the inputs include user set U and user-item rating matrix R. While, the output is T, which denotes the hierarchical user criteria tree. Furthermore, $c_i.rating$ denotes the rating matrix for each cluster, and M_{i*j} denotes the entry of the mutual-information-based distance proximity matrix. $M_{|U|*|U|}$ is symmetric and the diagonal is zero.

In Algorithm 1, Line 1 initialises the leaf nodes of the user criteria cluster tree T by assigning each user into a cluster. Lines 2 - 8 aim to compute mutual-information-based distances among clusters. In Line 9, the closest pair of clusters

Algorithm 1 The User Criteria Clustering Analysis

Input: U, R
Output: T
1: $K_C = K_T = |U|$, $c_i.ID = i$, $c_i \leftarrow u_i$, $c_i.rating \leftarrow R_{u_i}$, $T.Node_i \leftarrow c_i$, $T.Node_i.left = T.Node_i.right = Null$
2: **while** $K_C > 1$ **do**
3: **for** $\forall c_i \in C$ **do**
4: **for** $\forall c_j \in C \wedge i \neq j$ **do**
5: $M_{i*j} = D(c_i.rating, c_j.rating)$
6: $(c_i, c_j) \leftarrow argmin(D(c_i.rating, c_j.rating))$
7: **end for**
8: **end for**
9: $K_C = K_C - 1$, $K_T = K_T + 1$, $c_{temp} \leftarrow merge(c_i, c_j)$, $c_{temp}.U \leftarrow merge(c_i.U, c_j.U)$, $T.Node_{K_T} \leftarrow c_{temp}$, $T.Node_{K_T}.U \leftarrow c_{temp}.U$, $T.Node_{K_T}.left \leftarrow c_i$, $T.Node_{K_T}.right \leftarrow c_j$, $T.Node_{c_i.ID}.parent = T.Node_{c_j.ID}.parent = T.Node_{K_T}$, $c_i \leftarrow c_{temp}$, $C.remove(c_j)$, $c_i.ID = K_T$
10: **end while**
11: **return** T

are merged as a new cluster c_{temp}, while, a new cluster user set $c_{temp}.U$. c_{temp} and $c_{temp}.U$ are assigned as the latest internal node $T.Node_{K_T}$ and $T.Node_{K_T}.U$, respectively $T.Node_{c_i.ID}$ and $T.Node_{c_j.ID}$ are assigned as the left/right child node of the $T.Node_{K_T}$. Meanwhile, $T.Node_{K_T}$ becomes the parent node for these two nodes.

3.2 Object Community

Recall that, in our approach, each item with a particular feedback rating is regarded as **an object**. Mathematically, an **Object Community** OC is a subgraph of a preference network, which can be defined as a three-tuple, i.e., $OC =< U, O, E >$, where $OC.U \leftarrow T.Node.U$, $OC.O \leftarrow T.Node.parent.O$ and $OC.E = \{(u_i, o^{\tau_k}_{item_j})|u_i \in OC.U, o^{\tau_k}_{item_j} \in OC.O\}$. The edge between user u_i and object $o^{\tau_k}_{item_j}$ is represented as $e_{(u_i, o^{\tau_k}_j)}$. The weight of edge $w_{o^{\tau_k}_{item_j}}$ is formulated using Equation 4, where $\deg(o^{\tau_k}_{item_j})$ denotes the degree of the corresponding object vertex.

$$w_{o^{\tau_k}_{item_j}} = \frac{1}{\deg(o^{\tau_k}_{item_j})} \quad (4)$$

The object communities are formed by leveraging network modularity approach [11]. Traditionally, modularity method starts off with each vertex representing a community which contains only one member, and then it calculates the changes of modularity to choose the largest of them [12]. However, in traditional method, the order of objects dramatically affects the computation time and efficiency [13]. In order to alleviate this issue, in our approach, we calculate the distance value $dv(o^{\tau_k}_{item_j})$ for each object $o^{\tau_k}_{item_j}$ belonging to the object community of its parent node $T.Node.parent.O$ by using Equation 5 and 6.

$$dv(o^{\tau_k}_{item_j}) = (\sum_{u_i \in T.Node.parent.U}^{T.Node.parent} e_{(u_i, o^{\tau_k}_{item_j})} - \sum_{u_i \in T.Node.U}^{T.Node} e_{(u_i, o^{\tau_k}_{item_j})}) \times w_{o^{\tau_k}_{item_j}} \quad (5)$$

$$e_{(u_i,o^{\tau_k}_{item_j})} = \begin{cases} 1 & \text{if the rating value which user } u_i \text{ gives item } item_j \text{ equals to } \tau_k \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Based on the decreasing order of the $dv(o^{\tau_k}_{item_j})$, the **Modularity Gain** $\Delta Q_{o^{\tau_k}_{item_j}}$ is formulated in Equation 7, where the notations are explained as follows:

- m : The sum of the weights of all the edges in CG
- $m_{CG}(o^{\tau_x}_{item_j})$: The weight sum of the edge set $\{e_{(u_i,o^{\tau_k}_{item_j})}|o^{\tau_k}_{item_j} \in CG.O \setminus (u_i, o^{\tau_x}_{item_j}) \wedge u_i \in CG.U\}$
- $m_{OC}(o^{\tau_x}_{item_j})$: The weight sum of the edge set $\{e_{(u_i,o^{\tau_k}_{item_j})}|o^{\tau_k}_{item_j} \in CG.O \setminus (u_i, o^{\tau_x}_{item_j}) \wedge u_i \in OC.U\}$
- $l_{CG}(o^{\tau_k}_{item_j})$: The weight sum of the edge set $\{e_{(u_i,o^{\tau_k}_{item_j})}|u_i \in CG.U\}$
- $l_{OC}(o^{\tau_k}_{item_j})$: The weight sum of the edge set $\{e_{(u_i,o^{\tau_k}_{item_j})}|u_i \in OC.U\}$

$$\begin{aligned} \Delta Q_{o^{\tau_k}_{item_j}} =& [\frac{m_{OC}(o^{\tau_x}_{item_j}) + 2 \cdot l_{OC}(o^{\tau_k}_{item_j})}{2m} - (\frac{m_{CG}(o^{\tau_x}_{item_j}) + l_{CG}(o^{\tau_k}_{item_j})}{2m})^2] \\ & - [\frac{m_{OC}(o^{\tau_x}_{item_j})}{2m} - (\frac{m_{CG}(o^{\tau_x}_{item_j})}{2m})^2 - (\frac{l_{CG}(o^{\tau_k}_{item_j})}{2m})^2] \end{aligned} \tag{7}$$

If $\Delta Q_{o^{\tau_k}_{item_j}}$ is positive, object $o^{\tau_k}_{item_j}$ is added into the object community of the current node of the tree $T.Node.O$ for which its gain is maximum. Otherwise, $o^{\tau_k}_{item_j}$ only stays in $T.Node.parent.O$. The modularity discrepancy $\Delta Q_{o^{\tau_k}_{item_j}}$ is expected to be as large as possible, so that $item_j$ is more likely to be rated as τ_k by users in the user community of the current node of the user criteria tree $T.Node.U$ than those outside the community. Furthermore, some objects are connected with limited users. If it is randomly distributed, these objects will be removed from higher object communities. On the other hand, such objects may be connected with particular user groups. Therefore, it is always being maintained in some object communities.

The hierarchical object community generation algorithm is demonstrated in Algorithm 2. Lines 1-11 aim to initialise the top object community based on the user criteria clustering tree. Lines 12-31 tend to generate the object community OC for each node of the user criteria cluster tree T. The output of the algorithm is the object community OC set, and each OC is assigned to the related node of the user criteria clustering tree T.

3.3 Facet Object Set

In the previous two steps, both user and object community are supposed to be figured out for each node of the user criteria clustering tree. The user community

Algorithm 2 The Hierarchical Object Community Generation Algorithm

Input: T, $CG = < U, O, E >$
Output: T, $\{OC\}$

$index = T.Node.size() - 1$
$OC_{index}.U \leftarrow T.Node_{index}.U$
$tempO1 \leftarrow CG.O$
for $\forall o_j^{\tau x} \in tempO$ **do**
  for $\forall u_i \in OC_{index}.U$ **do**
    $sum = sum + e_{(u_i, o_j^{\tau x})}$
  end for
  if $(sum == 0)$ **then**
    $tempO1.remove(o_j^{\tau x})$
  end if
end for
$OC_{index}.O \leftarrow tempO1, T.Node_{index}.O \leftarrow OC_{index}.O$
for $\forall T.Node_{index} \in T \wedge T.Node_{index} \neq Null$ **do**
  $tempO2 \leftarrow T.Node_{index}.parent.O$
  for $\forall o_j^{\tau x} \in tempO2$ **do**
    $tP = tC = 0$
    for $\forall u_i \in T.Node_{index}.parent.U$ **do**
      $tP = tP + e_{(u_i, o_j^{\tau x})}$
    end for
    for $\forall u_i \in T.Node_{index}.U$ **do**
      $tC = tC + e_{(u_i, o_j^{\tau x})}$
    end for
    $distanceValue_{o_j^{\tau x}} = (tP - tC) * w_{o_j^{\tau x}}$
    $distanceQue[].add(distanceValue_{o_j^{\tau x}})$, $sort(distanceQue[])$
  end for
  for $\forall distanceValue_{o_j^{\tau k}} \in distanceQue[]$ **do**
    calculate $\Delta Q_{o_j^{\tau k}}$
    if $(\Delta Q_{o_j^{\tau k}} < 0)$ **then**
      tempO2.remove($o_{item_j}^{\tau k}$)
    end if
  end for
  $OC_{index}.O \leftarrow tempO2$, $T.Node_{index}.O \leftarrow OC_{index}.O$, $OC_{index}.U \leftarrow T.Node_{index}.U$
end for
return OC, T

shares a common preference and accepts a similar criterion of items. Hence, the object community of this level implies a particular facet of the real-world. One important feature for hierarchical object community is that the lower level, the more significant correlations among objects. Too low levels of object community cannot include all the relevant objects. While, too high levels of object community may consist of too much noisy objects. Therefore, we narrow the scope of the object community to generate the corresponding facet object set, which implies the preference of a certain user community.

Let $FO = \{o_i | o_i \in O\}$ denote the facet object. The objects in a particular facet object are not only correlated with others, but also evaluated under the same criteria by a group of users. In terms of each internal node $T.Node$ with child nodes $T.Node.left/T.Node.right$, users in the user community of left child node $T.Node.left.U$ also have interactions with part of objects belonging to the object community of right child node $T.Node.right.O$, and vice versa. Equation

8 defines the distance between two child nodes of current internal node. The community distance value $CDist(T.Node)$ is smaller if two objects in object communities of child nodes are more frequently and evenly connected with users in both two child user communities. It is necessary to specify a minimum acceptable threshold value, i.e., δ. If $CDist(T.Node) \geq \delta$, the contraction of facet object set will be terminated.

$$CDist(T.Node) = \sqrt{\sum_{o_{item_j}^{\tau_k} \in T.Node.O}^{T.Node} \left(\frac{\frac{\sum_{u_i \in T.Node.left.U}^{T.Node.left.U} e_{(u_i, o_{item_j}^{\tau_k})}}{|T.Node.left.U|}}{\frac{\sum_{u_i \in T.Node.U}^{T.Node.U} e_{(u_i, o_{item_j}^{\tau_k})}}{|T.Node.U|}} - \frac{\frac{\sum_{u_i \in T.Node.right.U}^{T.Node.right.U} e_{(u_i, o_{item_j}^{\tau_k})}}{|T.Node.right.U|}}{\frac{\sum_{u_i \in T.Node.U}^{T.Node.U} e_{(u_i, o_{item_j}^{\tau_k})}}{|T.Node.U|}} \right)^2} \tag{8}$$

3.4 Context Specific Inter-Personalised Trust Calculation

In terms of the inquired item $IE.item_j$ in particular enquirer, more than one facet object sets normally exist. Therefore, in order to make a more accurate prediction for enquirer $IE.u_i$, the system tends to compare the user's previous interaction records with the particular facet object sets related to inquired item, and then figure out the most trustable facet object set. Finally, the system suggests the most trustable item to the user. In our approach, the context specific inter-personalised trust value for particular facet object set is mainly determined by two factors: **Distance** and **Support**.

Distance represents the divergence between user's preference, R_{u_i} and facet object set, FO. It can be calculated by using Equation 9.

$$Dist(u_i, FO_j) = \sqrt{\sum_{R_{u_i}.r_{i,k} \neq 0, o_k^{\tau_x} \in FO_j}^{R_{u_i}, FO_j} \left(\frac{R_{u_i}.r_{i,k} - \tau_x}{|u_i.ratedItemSet \cap FO_j.ItemSet|} \right)^2} \tag{9}$$

In Equation 9, $|u_i.ratedItemSet \cap FO_j.ItemSet|$ denotes the number of items in facet object set $FO_j.ItemSet$ which are rated by u_i. While, $(R_{u_i}.r_{i,k} - \tau_x)$ calculates the difference between the rating given by u_i and τ_x implied by the object $o_{item_k}^{\tau_x}$ in facet object FO_j. $Dist(u_i, FO_j)$ is supposed to be small if objects in the facet object set FO_j are more appropriate for user's criteria about inquiry $item_k$.

Support is the ratio that each facet object set FO_j supports the rating history of user u_i, which is formulated in Equation 10.

$$Support(u_i, FO_j) = \frac{|u_i.ratedItemSet \cap FO_j.ItemSet|}{|u_i.ratedItemSet \cup FO_j.ItemSet|} \tag{10}$$

By considering both distance and support, the context specific inter-personalised trust value is formulated in Equation 11.

$$Trust(u_i, FO_j) = \frac{Support(u_i, FO_j)}{Dist(u_i, FO_j)} \tag{11}$$

4 Experiments

Experiments are conducted to analyse the performance of the community-based trust estimation approach. In the experiments, we compare the proposed approach with two memory-based collaborative filtering approaches, i.e. the user-based approach and the item-based approach, and one traditional data mining algorithms, i.e. K-Nearest Neighbour algorithm (KNN) [14].

4.1 Data Set

The real-world public dataset collected by Paolo Massa has been used for the experiments [15]. The dataset was crawled from *epinions* [1], which is a general consumer review website allowing users to share comments and reviews regarding various items, such as cars, books, music, etc. The ratings for each item ranges from 1 to 5. The dataset contains 195 users, 200 items and 5035 reviews.

A realistic collaborative filtering matrix probably contains millions of users and items. In practice, users give ratings to a few items only, and this results in a sparse matrix. The "sparseness" of a collaborative filtering matrix is defined as the percentage of empty cells [15]. Figure 2 demonstrates the number of users who created reviews. The X axis in Fig. 2 represents the user ID, while the Y axis indicates the item rating amount of the corresponding user. The sparseness of the dataset is around 87.1%, and more than 17% users rate no more than five items. The mean number of reviews is 25.82 with a standard deviation of 24.40, and the median is 19.

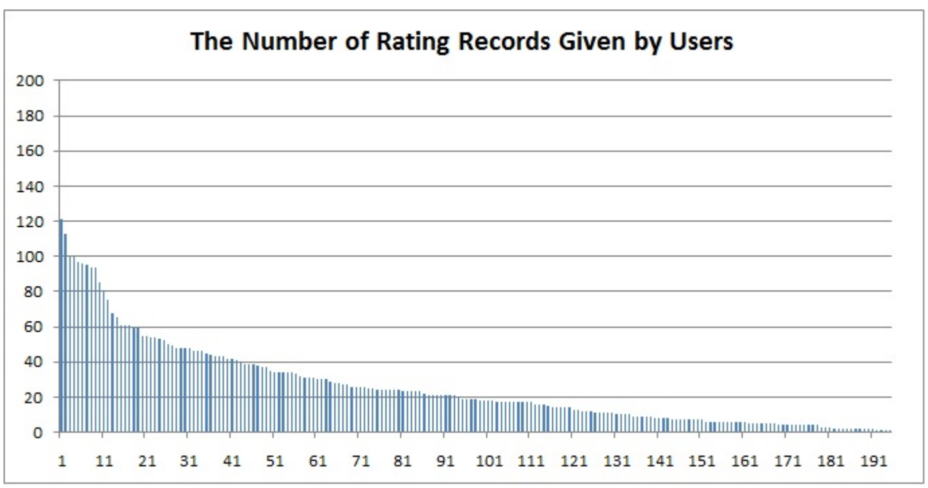

Fig. 2: Numbers of Reviews Rated by Users with Cold Start Users

[1] www.epinions.com/

4.2 Experimental Results

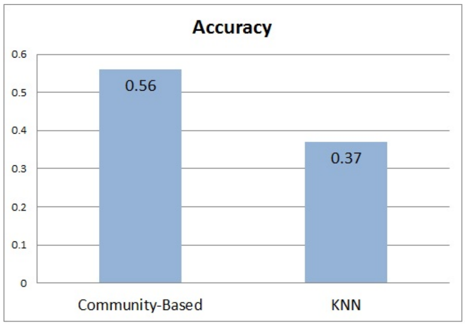

Fig. 3: Accuracy for Existing Users

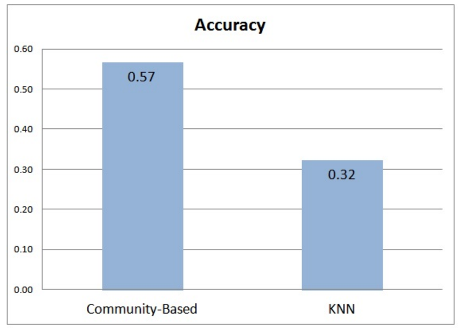

Fig. 4: Accuracy for Cold Start Users

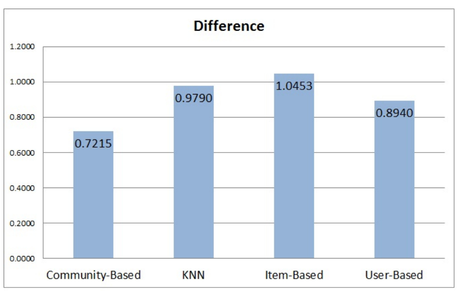

Fig. 5: Difference for Existing Users

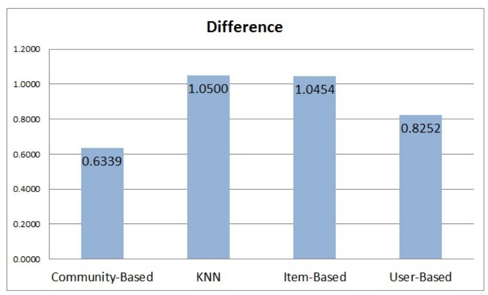

Fig. 6: Difference for Cold Start Users

When a new user enters the system without any rating history, it is difficult to predict his or her preference since the user has never given any ratings before. We consider users with less than five rating records as "cold start users" [16]. The traditional collaborative filtering algorithms are usually unable to provide high quality recommendations for this group of users. Moreover, accurate predictions also create an incentive for such users to continue using the system. Therefore, we also compare algorithms' performances for "cold start users".

In the experiment, we mainly use two metrics, i.e., accuracy and difference, to compare the performance of the community-based trust estimation algorithm and the other three algorithms. Specifically, the accuracy signifies the percentage of potential quality prediction of items which are equal to the actual feedback rating values given by enquirers. However, neither user-based nor item-based approach can predict the exact rating values for required items. Hence, difference is adopted as another comparison metric. It measures the average distance between the actual and predicted rating values.

Comparison of Accuracy. Figure 3 illustrates the accuracy comparison of both the community-based approach and the KNN. The accuracy of proposed algorithm reaches 0.56, which is much higher than that of the KNN at 0.37. In Figure 4, we compare the algorithms for the cold start users, the accuracy of community-based algorithm increases by 0.01, which is significantly higher than KNN. In this sense, for a new user without enough rating records, the community-based recommendation algorithm is still capable of providing trustable suggestions to users.

Comparison of Difference. Figure 5 compares the difference values of the four algorithms. The community-based algorithm performs better than the other three algorithms, where the difference is approximate 0.72. Furthermore, as can be seen from Figure 6 that, the difference of the community-based approach narrows to 0.6339 in terms of the "cold start users". However, the difference of the KNN and the item-based algorithm increased to above 1. Although the performance of the user-based recommendation algorithm performs better than KNN and item-based approach, the difference (0.8252) is still higher than the community-based trust estimation algorithm.

5 Conclusion

In this paper, we proposed a community-based trust estimation approach to mine context specific inter-personalised trust in preference networks. In the approach, we organise the preference network as a set of more manageable interrelated communities. The approach mainly focuses on users with similar preference, and groups them into various user communities. Furthermore, object communities are partitioned to imply the interest and criterion of user communities for particular items. Finally, distance and support are considered in the approach to ascertain the most confident facet object set, and make the trustable quality prediction based on the rating value of the object about the enquired item $IE.item_j$ in this facet object set. From the experimental results, it can be seen that the community-based approach gives better performance than some other approaches in terms of both difference and accuracy, even under the "cold start users" situation. However, the community-based approach manages trust information in a centralized manner. In the future, we will extend the community-based mechanism to distributed environments.

References

1. Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.
2. Georgios Pitsilis and Lindsay Marshall. *Trust as a key to improving recommendation systems*. Springer, 2005.
3. Lipi Shah, Hetal Gaudani, and Prem Balani. Survey on recommendation system. *system*, 137(7), 2016.
4. Heng Luo, Changyong Niu, Ruimin Shen, and Carsten Ullrich. A collaborative filtering framework based on both local user similarity and global user similarity. *Machine Learning*, 72(3):231–245, 2008.
5. Dongsheng Li, Chao Chen, Qin Lv, Li Shang, Yingying Zhao, Tun Lu, and Ning Gu. An algorithm for efficient privacy-preserving item-based collaborative filtering. *Future Generation Computer Systems*, 55:311–320, 2016.
6. Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.
7. Roger Guimerà, Marta Sales-Pardo, and Luís A Nunes Amaral. Module identification in bipartite and directed networks. *Physical Review E*, 76(3):036102, 2007.
8. Jian-Guo Liu, Tao Zhou, Hong-An Che, Bing-Hong Wang, and Yi-Cheng Zhang. Effects of high-order correlations on personalized recommendations for bipartite networks. *Physica A: Statistical Mechanics and its Applications*, 389(4):881–886, 2010.
9. Xiaobo Zhou, Xiaodong Wang, Edward R Dougherty, Daniel Russ, and Edward Suh. Gene clustering based on clusterwide mutual information. *Journal of Computational Biology*, 11(1):147–161, 2004.
10. Zaher Dawy, Joachim Hagenauer, Pavol Hanus, and Jakob C Mueller. Mutual information based distance measures for classification and content recognition with applications to genetics. In *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, volume 2, pages 820–824. IEEE, 2005.
11. Nicole Belinda Dillen and Aruna Chakraborty. Modularity-based community detection in fuzzy granular social networks. In *Proceedings of the International Congress on Information and Communication Technology*, pages 577–585. Springer, 2016.
12. Zhongying Zhao, Shengzhong Feng, Qiang Wang, Joshua Zhexue Huang, Graham J Williams, and Jianping Fan. Topic oriented community detection through social objects and link analysis in social networks. *Knowledge-Based Systems*, 26:164–173, 2012.
13. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
14. Daniel T Larose. k-nearest neighbor algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining*, pages 90–106, 2005.
15. Paolo Massa and Bobby Bhattacharjee. Using trust in recommender systems: an experimental analysis. In *Trust Management*, pages 221–235. Springer, 2004.
16. Wayne Xin Zhao, Sui Li, Yulan He, Edward Y Chang, Ji-Rong Wen, and Xiaoming Li. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. *IEEE Transactions on Knowledge and Data Engineering*, 28(5):1147–1159, 2016.