A New Similarity Measure Based on Item Proximity and Closeness for Collaborative Filtering Recommendation

Sama Jamalzehi *, Mohammad Bagher Menhaj

Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Department of Electrical Engineering Amirkabir University of Technology, Tehran, Iran

Abstract

Recommender systems utilize information retrieval and machine learning techniques for filtering information and can predict whether a user would like an unseen item. User similarity measurement plays an important role in collaborative filtering based recommender systems. In order to improve accuracy of traditional user based collaborative filtering techniques under new user cold-start problem and sparse data conditions, this paper makes some contributions. Firstly, we provide an exposition of all-distance sketch (*ADS*) node labelling which is an efficient algorithm for estimating distance distributions; also we show how the *ADS* node labels can support the approximation of shortest path (*SP*) distance. Secondly, we extract items' features and accordingly we describe an item proximity measurement using ochiai coefficient. Third, we define an estimation of closeness similarity, a natural measure that compares two items based on the similarity of their features and their rating correlations to all other items, then we describe our user similarity model. Finally, we show the effectiveness of collaborative filtering recommendation based on the proposed similarity measure on two datasets of MovieLens and FilmTrust, compared to state-of-the-art methods.

Keywords: collaborative filtering, recommender system, user similarity, Closeness similarity, All-distance sketch.

1. Introduction

Recommendation System (RS) as a type of information filtering system have been successfully developed to produce useful data. Collaborative Filtering (CF) is the most widely used technique in recommender systems to provide personalized suggestion. The main advantage of CF is that it recommends unconventional items to an active user by analyzing rating information of the other users in his/her neighborhoods [1].

CF algorithms are categorized into two classes, model based and neighborhood based [1]. Model based algorithms define the implicit similarity by learning a model from the training data and often give very little intuition of the people's preferences. While neighborhood based algorithms create a prediction for an active user by finding his/her most similar neighbors. After finding a neighborhood of similar users, different methods are applied to integrate preferences of neighbors to make a prediction for an active user for a product that he/she has not rated [2]. Most of the electronic commercial systems employed

* Corresponding author. Email: s.jamalzehi@qiau.ac.ir

neighborhood based recommender systems to make personalized suggestion, as these systems are intuitive and relatively simple to implement.

Generally, the essential part of CF algorithms is to use proper metrics for measuring the similarity between each two users [2]. Local similarity measures, such as Pearson similarity measure [3] and Cosine similarity measure [2] that are based on the similarity estimation between two users through the set of common items rated by both users, take into consideration only the immediate neighborhoods; However, global measures can assign meaningful similarity scores to those pairs that are more than two hops apart. Note that as these measures are often computationally more expensive, it is hard to apply them to graphs with tens to hundreds of millions of nodes [4].

Aiming for accuracy, we develop a new model based on the combination of local information of ratings and global properties of rated items. Our approach consists of two key steps. In the first, we describe all-distances sketch (*ADS*) labels through a sketching algorithm that assigns a label to each node in the graph. We demonstrate how *ADS* labels can be developed for estimating the shortest path(s) between two given nodes. The *ADS* labels were initially developed for estimating the number of nodes reachable from a given node [5]. An efficient advantage of the *ADSs* over the Thorup-Zwick outline is that they are useful for distance estimations, closeness similarity metrics and neighborhood sizes [6], [7].Based on the previous studies, we show that assigning *ADS* label to each node in a graph can be done efficiently, with a logarithmic total number of edge traversals.

In second, we extract items' features from the relevant database to create feature vector for every user rated item. Accordingly, we use of ochiaiindex to define a synthetic factor for measuring proximity between two items based on their feature vectors. In the third step, we first create undirected item-item graph from the user-item rating matrix, then we define an estimator of the closeness similarity between items, where *ADS* node labels and proximity levels are considered to distance estimation amongst all items. After that, we present a novel model for finding similarity between a pair of users in which the proposed closeness estimator is utilized for measuring similarity between each pair of users' rated items.

Lastly, we show the effectiveness of our similarity measure through a large-scale experimental study on two benchmark movie datasets of movies with different scales and sparsity levels (*Movie Lens* and *Film Trust*). The experimental results show that the proposed model produces more accurate recommendations in terms of MAE, when compared to the traditional similarity measures.

2. Background and Related Works

The most widely used techniques in recommendation systems are neighborhood based collaborative filtering algorithms, in which similarity computation between items or users is the most critical step. For user-based CF algorithms, there are many different methods to compute similarity between users. In this section, we first analyze the most important existing similarity measures along with their limitations. Then, we present the motivation of the proposed similarity model.

2.1. Similarity Measures in User-Based CF

In most of traditional user-based CFs, the similarity values between users are computed based on Pearson correlation coefficient (PCC) [3] and Cosine [2] measures. The PCC measures how two users are linearly correlated to each other. However, it only considers the absolute rating values on co-rated items, while the number of co-rated items is also important for measuring similarity between two users. The cosine similarity between two users and v is measured by computing the cosine of the angle between rating

vectors of u and v, even though it does not consider the users' preferences with various rating scales. The Jaccard similarity [8] is another commonly used similarity measure, but its drawback is that it only considers the number of common ratings between two users. The formulas of the above mentioned similarity measures are defined as follows:

$$sim(u,v)^{PCC} = \frac{\sum_{p\in I}(r_{u,p}-\overline{r_u})(r_{v,p}-\overline{r_v})}{\sqrt{\sum_{p\in I}(r_{u,p}-\overline{r_u})^2}\cdot\sqrt{\sum_{p\in I}(r_{v,p}-\overline{r_v})^2}} \quad (1)$$

$$sim(u,v)^{CPCC} = \frac{\sum_{p\in I}(r_{u,p}-r_{med})(r_{v,p}-r_{med})}{\sqrt{\sum_{p\in I}(r_{u,p}-r_{med})^2}\cdot\sqrt{\sum_{p\in I}(r_{v,p}-r_{med})^2}} \quad (2)$$

$$sim(u,v)^{COS} = \frac{\vec{r_u}.\vec{r_v}}{\|\vec{r_u}\|.\|\vec{r_v}\|} \quad (3)$$

$$sim(u,v)^{Jaccard} = \frac{|I_u|\bigcap|I_v|}{|I_u|\bigcup|I_v|} \quad (4)$$

Where the corresponding parameters are defined as follows. $\bar{r}_u$ and $\bar{r}_v$ are the average rating values of user *u* and *v* respectively. $r_{u,p}$ and $r_{v,p}$ denote the ratings of item *p* by user *u* and user *v* respectively. $\vec{r}_u$ and $\vec{r}_v$ are the vectors of user *u* and user *v* rated, respectively. r_{med} is medium rate in the rating scale (For considering the impact of positive and negative ratings, the all rates that are greater than r_{med} were assumed positive and others were assumed negative ratings). I_u, I_v and I represent the set of rating items for users *u* and *v*, and the set of co-rated items that are rated by both users *u* and *v*, respectively.

As these similarity measures have some weaknesses such as data sparsity, new user cold-start and scalability, many improved similarity measures have been introduced to overcome these drawbacks. The Mean Square Distance (MSD) is another measure [9] that only considers the absolute ratings. For incorporating the ratio of common ratings into MSD measure, it has been combined with Jaccard measure, called JMSD measure [10].The heuristic PIP measure [11] is the most recently used similarity measure, which consists of three factors of similarity, Proximity, Impact and Popularity. The proximity factor takes an absolute reference like as median of the rating scale to consider whether two ratings are in agreement or not. The impact factor exhibits how strongly an item is liked or disliked by users. Note that when ratings are not in the same direction of median, the computation of proximity and impact will be repeatedly penalized. The popularity factor solve this problem by giving more importance to a rating that is far away from the item's average rating. This factor presents how two ratings are different with other ratings. Although the PIP measure can provide successful results, it not considers the global information of ratings and the proportion of common ratings.

Bobadillla et al. [12] combined basic measures to introduce a new similarity measure named Mean-Jaccard-Difference (MJD), in which the information of numerical ratings are used as well as the distributions of user ratings. However, it also suffers from few co-rated items problem. Haifengliu et al. [13] produced an improved heuristic similarity model called NHSM to alleviate the drawbacks of initial PIP based measure. They picked up a non-linear formula to calculate similarity measure based on three factors of proximity, significance and singularity. However, in user similarity computation with NHSM measure, only co-rated items are considered.

As the already measures only consider the co-rated items in similarity calculation between two users, Be et al. addressed this problem by introducing two similarity measures based on Bhattacharyya Coefficient, BCF_{med} and BCF_{cor}, which utilize all rating data in user similarity measurement [14]. The main challenge of the BCF measures is that they ignore differences in two users' opinions on co-rated items. Moreover, these measures unable to compute user similarity when each of two user's ratings on every rated item have same distances from the item's

median rating (in BCF_{med}) or the item's average rating (in BCF_{cor}).

2.2. The Motivation of New Similarity Estimation Model

While several similarity measures have been introduced to overcome some limitations of the traditional similarity techniques, they still have some drawbacks. The contributions in this paper are related to alleviate the following drawbacks of similarity measures.

• The correlation based measures that utilize just co-rated items while computing similarity between two users, are not suitable under the sparsity condition where the number of individual user ratings is less and number of co-rated items is few or none.

• Ignoring the global information about the user's preferences usually leads to low accurate predictions.

• High pair wise similarity report between two different users who have rated the same item, despite they may hold different opinions on it.

• Discarding the pure rating values will become difficult to discriminate a many users with different item ratings, thus it leads to very low accurate similarity measurement.

3. Opening Remarks For Distance Estimation

In this section, we first provide a brief study from [6] and [7] about all distance sketch (ADS) labeling. Then, we explore how ADS labels can be used for shortest path estimation in a graph.

3.1. All-Distance Sketch Labelling Review

In this paper, we consider undirected item-item graph. For two nodes v and u, d_{vu} and π_{vu} indicate the shortest-path (*SP*) distance from v to u, and dijkstra rank of u with respect to v, respectively. The π_{vu} is defined as u's position in the list of nodes sorted by increasing distance from v. For two nodes u and v, the $\Phi_u(v)$ is used for the set of nodes j that are within a distance from u to v ($\pi_{vj} \leq \pi_{vu}$). For $d \geq 0$ and node v, $N<d(v)$ is the set of nodes that are of distance less than d from v(the <d neighborhood of v). For a numeric function r: $X \rightarrow [0,1]$ over a set X, the function $k_r^{th}(X)$ gives back the k-th lowest value in the range of r on X. If $|X|<k$, then $k_r^{th}(X) = 1$.The all-distance sketch (*ADS*) labels are defined with respect to a random rank assignment to nodes such that for any u, $rd(u) \sim U[0,1]$. It is supposed that each *ADS* contains a node and a distance, such:

$$ADS(v) = \{(u, d_{vu}) \mid rd(u) < k_r^{th}(\Phi_{<u}(v))\} \quad (5)$$

where $\Phi<u(v)$ indicates the set of nodes that are closer to v than u.

Specifically, a node u appertains to *ADS* (v) if u is between the k nodes with smallest rank r in the sphere of radius d_{vu} around v. The maximum expected size of *ADS*(v) is *k. Ln (n)*, where n is the number of nodes reachable from v. For a node $u \in ADS\ (v)$, the k-th smallest rank value amongst nodes that are closer to v than u, is defined as follow:

$$p_{vu} = k_r^{th}(\Phi_{<u}(v)) \quad (6)$$

Where, $k_r^{th}(\Phi_{<u}(v)) = k_r^{th}(\{i \in ADS(v) \mid d_{vi} < d_{vu}\})$.

Another practical function is threshold rank, the maximum rank value of every node at distance x from v to be included in *ADS* (v) that is defined as:

$$\tau_v(x) = k^{th}(N_{<x}(v)) \quad (7)$$

So if node u is included in *ADS (v)* then $p_{vu} = \tau_v(d_{vu})$ We also use of the following inverse function to gain a lower bound on the distance d_{vi} for identifying all nodes i that not belong to *ADS (v)*.

$$\tau_v^{-1}(z) = \max\{d_{vi} \mid k_r^{th}(\Phi_{<i}(v)) > z\} \quad (8)$$

3.2. Shortest Path Calculation

Node labels have been used efficiently for shortest path calculation in road networks [15] and medium-size un weighted social graphs [16]; However, these strict labels are much more expensive to compute than *ADSs*. Based on the previous studies, we demonstrate how the use of *ADS* distance labels are efficient to shortest path estimation. We can use *ADS(v)* and *ADS(u)* as 2-hop labels to obtain a good estimate of the shortest distance d_{vu} as below:

$$d_{vu} = \min\{d_{vi} + d_{ui} \mid i \in ADS(v) \cap ADS(u)\} \quad (9)$$

In order to obtain a good estimate, we have to select a proper node *i* that belongs to intersection of *ADSs*. If *i* is *k*-th within the intersection of *ADSs*, then the sufficient condition for it to be within the random permutation produced on intermediate nodes and the nodes $\Phi v(i) \cup \Phi u(i)$ is satisfied. This can happen with probability of min $\{1, \frac{k}{|\Phi_u(i) \cup \Phi_v(i)|}\}$ [7].

Dijkstra algorithm is often used for calculating shortest path in a graph, the best case running time of this algorithm is *O (m + n log n)*, where *n* and *m* are the number of users in social network and the number of relations between users, respectively. While this time complexity can be significantly decreased with only one computation of *ADSs*.

If the size of largest set of *ADSs* to be considered as S (which is numerically very small), query time of distance estimation between two nodes will be equal to *O(S log S)*. The value of S depends on the value of two parameters n and k in *ADS's* computation algorithm like it takes *O(k log n)* time to compute in the worst case.

Based on the above explanations, query time of distance estimation with using *ADSs* is *(O (k log (n)× log (k log(n)))*. Since we are usually considered small value of k (3 in here), therefore shortest path distance can be estimated in *(O(log (n) × log (log(n)))* time that is more efficient than time complexity of Dijkstra algorithm [6].

4. IPFE: An Item Proximity Measure Based on Feature Extraction

In this section, we introduce a measure of item proximity (*IP*) that is used in fifth, as a factor in item closeness estimation. Initially, we extract the items information by automatic indexing, which is a typical feature extraction function for text documents [17]. Then we create the desired items' feature vector for measuring proximity between each pair of items. Indeed, we present every item u as an item's feature vector $X_u = \{X_{u1}, X_{u2}, \ldots, X_{ut}\}$ in the t-dimensional feature space. Lastly, given a pair of feature vectors X_i and X_j that describe two items u and v, the ochai index [18] can be applied to measure their proximity as follow:

$$IP_{u,v} = \frac{\sum_{i=1}^{k} (X_{ui} \cap X_{vi})}{t} \quad (10)$$

Where t is the number of elements in item feature vector. If the nth feature of i is equal to nth feature of j, $(X_{in} \cap X_{jn})$ is "1"; otherwise, it is "0".

5. The New Similarity Model Organization

The traditional similarity measures have obvious limitations, as mentioned in section 2. In this section, we first create an item graph from the rating matrix. Then we estimate the item closeness, which computes the similarity of two items based on their IP degree and a view of the whole graph. Finally, we introduce our user similarity measure.

5.1. Item-Item Graph Creation

We convert user-item rating matrix into item-item graph, in which nodes represent items and the value of weights on edges indicate the strength of correlations among items. For this purpose, we have employed the adjusted cosine (ACOS) measure [19] below a suggested threshold.

$$ACOS(v,u) = \frac{\sum_{i \in I}(r_{iv} - \overline{r_v})(r_{iu} - \overline{r_u}}{\sqrt{(r_{iv} - \overline{r_v})^2}.\sqrt{(r_{iu} - \overline{r_u}} \qquad (11)$$

Where I is the set of users rated both items v and u, r_{iv} is the rating made by user i on item v and $\overline{r_v}$ is the average rating of item v.

Every two items are linked together if their ACOS value is above a given threshold. For suggesting an appropriate threshold which be able to identify disconnected components (new cold items), the median absolute deviation (MAD) is used as a measure of dispersion, because it is a more robust estimator of rating scales than the sample variance or standard deviation [20].

Accordingly, in this work, an item graph is defined as an undirected weighted graph *G=(U, E),* where

- *U* is the node set (each item is regarded as a node in the graph *G*).
- *E* is the edge set. Associated with each edge $e_{v,u} \in E$, w_{vu} is a weight subject to $w_{vu}>0$, $w_{vu}=w_{uv}$.

$$w_{v,u} = \begin{cases} \dfrac{1}{ACOS(v,u)} & \text{if } ACOS(v,u) > threshold, \\ 0 & else. \end{cases} \qquad (12)$$

Where we use the poorly conservative threshold of median plus 2 times the MAD [20] to detect the minimal set(s) of outliers which should be pruned leaving the dataset.

5.2. Closeness Similarity Estimation

Using only the absolute value of common-rated items in similarity measurement between two users has obvious limitations, as mentioned in section 2. In this section, we estimate the closeness similarity, which computes the similarity of two rated items based on the overall view of rating matrix. More exactly, we consider the distance from each of these two users' rated items to all other items in the network, then based on the closeness's between two users' rated items, we measure how much these two users are similar in their interests.

In this study, the closeness similarity for all item pairs v and u is specified with the jaccard form [7], based on a distance decay function ρ and the shortest distance d_{vu}, as follow:

$$J(v,u) = \frac{\sum_{i \in ADS(v) \bigcap ADS(u)} \rho(\max\{d_{vi}, d_{ui}\})}{\sum_{i \in ADS(v) \bigcap ADS(u)} \rho(\min\{d_{vi}, d_{ui}\})} \qquad (13)$$

Where conditioned on mono tonicity of ρ, similarity is in [0,1].

The exact computation of closeness similarity have a high time complexity, because it requires two searches for finding the shortest path between each pair of nodes; However, a cost-effective estimation of closeness similarity can be derived using the item graph with *ADS* node labels. In this view, we can obtain reasonable results by settings $\rho(x)=1/1+x$, which gives us a global variant of Adamic-Adar (*AA*) measure [21], and $k=3$, as will be shown below. This choice of ρ leads incorporating only *IP* degree into "item–new cold item" similarity estimation. Generally, nevertheless, the distance function ρ can be any decay function such as Polynomial, Exponential or Gaussian, depending on the value of metric's flexibility.

In order to the formal computation of *ADSs*, we assign to each item v a normally distributed random rank $rd(v)$ with mean $\overline{r_v}$, the average rating value on item u. We compute the users' random ranks as below:

$$rd(v) = \sqrt{\sum_{i=1}^{n}(r_{iv} - \overline{r_v})^2} \qquad (14)$$

Where r_{iv} denotes the rating of item v by user i, and n is the total number of users who rated item v.

In the reminder we show how the values of $\rho(max\{d_{vi}, d_{ui}\})$ and $\rho(min\{d_{vi}, d_{ui}\})$ can be derived by good estimators. For this, we use αL* estimator of

Cohen [22] and U* estimator of Cohen [23] for estimating the distance functions $\rho(max\{d_{vi}, d_{ui}\})$ and $\rho(min\{d_{vi}, d_{ui}\})$, respectively, as these estimators are unique, monotone (non-increasing) and admissible (pare to variance optimal). Note that $\rho^{(\alpha L^*)}$ maximize and $\rho^{(U^*)}$ minimize the ρ estimate, therefore, the best possible scores of pair wise similarity can be accurately estimated by these estimators.

Lemma 5.1. The αL* estimate of $\rho(max\{d_{vi}, d_{ui}\})$ is

$$\rho^{\alpha L^*} = \begin{cases} 0 & \text{if } i \notin ADS(u) \cap ADS(v), \\ \dfrac{\alpha}{p_{\min}} \rho(\min\{d_{vi}, d_{ui}\}) & \text{if } i \in ADS(u) \cap ADS(v). \end{cases} \quad (15)$$

Where p_{min}= min$\{p_{vi},p_{ui}\}$, with p_{vi} and p_{ui} as defined in (2).

Proof. Since from [22], $\rho^{(\alpha L^*)}(\max\{d_{vi},d_{ui}\})=\alpha.\rho^{(L^*)}(\max\{d_{vi},d_{ui}\})$, there for we need to derive L^* estimator for estimating the maximum distance. In the case of $i \notin ADS(u) \cap ADS(v)$, $\rho^{(L^*)}$ is 0, because of there is no available information about maximum distance. In the other case, as the inclusion probability of node *i* is inversely proportional to its distance from u and v, the inverse probability estimate can be applied efficiently [7].

By applying the αL^* estimator with a rating independent choose of α, the mutual influence between two items which are far from each other, can be taken into account in the $\rho(max\{d_{vi},d_{ui}\})$ estimating. To do so, we pick α equal to two times the *IP* degree between two source nodes *v* and *u*, regardless of intermediate nodes *i*. With this setting of α, when IP_{vu}<0.5, αL^* estimator lies outside the ideal range on every outcome, when IP_{vu}=0.5, the estimator is equivalent to L^*(in this case the difference between the two item's ratings is ignored), and when IP_{vu}>0.5, the estimator lies the ideal range.

Lemma 5.2. The U^* estimate of ρ(min$\{d_{vi},d_{ui}\}$) with respect to any node $x \in X$, $(X \cap \{ADS(u) \cap ADS(v)\})$, conditioned on $\rho(d_{xi})=\rho(min\{d_{vi},d_{ui}\})$ and $i \in ADS(u) \cap ADS(v)$ is

$$\rho^{U^*} = \begin{cases} \inf\limits_{0 \le p_{xi} < p_{vi}} \dfrac{\rho(d_{xi}) - (p_{ui} - p_{vi})\dfrac{\rho(d_{ui})}{p_{ui}}}{p_{vi} - p_{xi}} & \text{if } p_{vi} \le p_{ui}, \\ \inf\limits_{0 \le p_{xi} < p_{ui}} \dfrac{\rho(d_{xi}) - (p_{vi} - p_{ui})\dfrac{\rho(d_{vi})}{p_{vi}}}{p_{ui} - p_{xi}} & \text{if } p_{ui} < p_{vi}. \end{cases} \quad (16)$$

and when $i \notin ADS(u) \cap ADS(v)$, $\rho^{(U^*}(min\{d_{vi}, d_{ui}\})$ is equal to 0.

Proof we apply an explicit construction of U^* estimator from Cohen [23]. Fixing random ranks on all nodes, the result depends on the threshold value $\tau_x(d_{xi})$, which is bottom-(*k*-1) smallest rank value of min$\{\Phi<i(v), \Phi<i(u)\}$. With this estimator, the tightest lower bound on ρ(min$\{d_{vi},d_{vi}\}$) can be obtained. This is the infimum of the function on all distances d_{xi} that are possible.

The first instance is $i \notin ADS(u) \cap ADS(v)$, node *i* is excluded from intersection of ADS(*u*) and ADS(*v*) if and only if *rd(i)* is greater than both of and $(k-1)_r^{th}(\Phi_{<i}(v))$.Specially, the rank of node *i* should not be smaller than the (*k*-1)-th smallest rank amongst nodes that are closer to *v* and *u* than *i*. When *rd(i)*~U[0,1], this happens with probability P_i, such $p_i = pr[rd(i) > (k-1)_r^{th}\{(\Phi_{<i}(u) \cap ADS(u)\} \cup (k-1)_r^{th}\{(\Phi_{<i}(v) \cap ADS(v)\}]$.Since $i \notin ADS(u)$ and $i \notin ADS(v)$, So *rd(i)* can not be *k*-th smallest rank in *ADS*(*u*) ∪*ADS*(*v*). On the other hand, we have not any boundaries (upper bound or lower bound) on the minimum distance. Therefore, the probability P_i is 0.

The next instance is when $i \in ADS(u) \cap ADS(v)$, the both of *ADS(u)* and *ADS(v)* contain *i* if and only if the rank of *i* is one of the *k* smallest ranks amongst nodes that are at least as close to *u* $(rd(i)< k_r^{th}(\Phi_{<i}(u)))$, and also close to *v*$(rd(i)< k_r^{th}(\Phi_{<i}(v)))$. We know that both distances d_{vi} and d_{ui}, and hence the inclusion thresholds p_{vi} and p_{ui}, can be computed from the intersection of *ADS(v)* and *ADS(u)* for all $i \in ADS(u) \cap ADS(v)$. Therefore, all of nodes *x* at distance smaller than

$\min\{d_{vi},d_{ui}\}$ from i can be found. Consequently, in this case we can estimate $\rho^{U^*}(\min\{d_{vi},d_{ui}\})$ properly. This gives us an overall estimate of minimum distance for all situations.

In the previous case, if *i* belongs to *ADS(u)* and not to *ADS(v)*, we should refer to condition $p_{vi} \le p_{ui}$, because this condition means that $p_{vi} < rd(i) \le p_{ui}$;We know that if $i \in ADS(u)$, then $p_{ui} = \tau_u(d_{ui})$. Furthermore, when $i \notin ADS(v)$, the inverse function $\tau_v^{-1}(rd(i))$ gives us a lower bound on distance d_{vi}. In this way, since we just know that the minimum distance is between $\tau_v^{-1}(r(i))$ and d_{ui}, so we set $\rho(d_{xi}) = \rho(d_{ui})$. Accordingly, the outcome of minimum distance estimator is $\rho^{U^*}(\min\{d_{vi}, d_{ui}\}) = \inf_{0 \le p_{xi} < p_{vi}} \frac{\rho(d_{ui})}{p_{ui}} - \frac{p_{vi}}{p_{xi}}$. The other situation in which node *i* belongs to *ADS(v)* and not to *ADS(u)*, conditioned on $p_{ui} < p_{vi}$, is symmetric.

After gathering all the information, we can estimate closeness similarity between each two items *v* and *u*, as below:

$$J^*(v,u) = \frac{\sum_{i \in ADS(v) \bigcap ADS(u)} \rho^{\alpha L^*}(\max\{d_{vi}, d_{ui}\})}{\sum_{i \in ADS(v) \bigcap ADS(u)} \rho^{U^*}(\min\{d_{vi}, d_{ui}\})} \quad (17)$$

Note that, the simple closeness similarity with α= 1, ignores the differences between two items, since it only considers the shorter path between rated items in the item graph. In order to improvethe accuracy of closeness similarity measure, we can employ the *IP* index to get an appropriate setting of the parameter α, as previously mentioned. This similarity measure is unbiased, because both of the estimators are unbiased.

5.3. ICCF: A Similarity Measure based on Item Closeness for Neighborhood-based CF

The proposed measure *(ICCF)* utilizes the above-mentioned item closeness estimator to compute similarity between each pair of users. Let I_X and I_Y be the two sets of items that have been rated by user *X* and *Y*, respectively. The similarity between the two users *X* and *Y* in *ICCF* metric is the function of closeness similarity between a pair of rated items (Eq. (14)).

$$ICCF(X,Y) = \sum_{x \in I_X} \sum_{y \in I_Y} J^*(x,y) \quad (18)$$

Now, we discuss some major properties of the proposed *ICCF* similarity measure.

- When there is no co-rated item between two users, *ICCF* measure can compute similarity between them, as it does not depend on number of co-rated items.
- In *ICCF* measure, the local and global informationsare considered based on correlation of the users' ratings in the item graph and item proximity (*IP*) values, respectively.
- In the condition where two users have rated different items, but the ratings created by users have similar distances from the mean rating, *ICCF* measure can compute two users' similarity efficiently.
- To improve the accuracy, *ICCF* similarity measure utilizes all of the user-item ratings in addition to their distances from the average item rating.

6. Experiments

This section presents the experimental procedure on two popular datasets.

6.1. Datasets

In this paper, two standard datasets are used in the experiments including *FilmTrust* and *MovieLens*. The FilmTrust dataset is a trust-based social network where users can rate movies. This dataset consists of 1986 users, 2071movies and 35,497 ratings. The rate values are numbers in the range of 0.5 to 4.0 with step 0.5.On the other hand, the MovieLens dataset was collected by the GroupLens research group and includes 100,000 ratings with 943persons and 1682 movies.

Each user in this dataset has rated at least 20 movies and he/she can assign numeric ratings to movies in the range from 1 to 5.

In this work, we have used content information about movies for computing item proximity (*IP*). We obtained the additional information about movie key features by crawling the internet movie database (www.imdb.com) include Actors/Actresses, Directors, Producers, Editors, Writers, Production companies.

6.2. Evaluation Metrics

In this paper to evaluate the recommendation methods, each of the two data sets are divided in to two parts, of which 80 % is taken as training set and remaining 20% as testing set. The *k*-nearest neighbors of users are computed using the training set, and then the predictions are generated based on the testing set with below equation.

$$P_{t,i} = \overline{R_t} + \frac{\sum_{x \in N_d(t)} sim(t,x).(R_{x,i} - \overline{R_x})}{\sum_{x \in N_d(t)} sim(t,x)} \quad (19)$$

Where $P_{t,i}$ indicates the predicted rating of the item '*i*' by the active user '*t*', $\overline{R_x}$ is the mean of user *x*'s ratings, sim(*t,x*) is the user similarity value between '*t*' and '*u*', $R_{x,i}$ represents the current rate of item *i* by user *x* and denotes the set of nodes of distance at most *d* from *t*.

There are many measures for evaluating prediction accuracy. These metrics are classified into accuracy metrics and coverage metrics [24]. In order to compare the accuracy of the proposed method with the other methods, we use of *MAE* (Mean Absolute Error) and *RMSE* (Root Mean Square Error) which are two most common measures of predictive accuracy.

$$MAE = \frac{\sum_{i=1}^{N} |r_i - p_i|}{N} \quad (20)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} |r_i - p_i|}{N}} \quad (21)$$

Where r_i and p_i are actual and predicted ratings of an item *i*, respectively, and *N* presents the total number of rates that are predicted by a recommender method.

6.3. Experimen Results

We will compare the results with different values of the number of nearest neighbors that is one of the most convential parameter to impact the performance of recommendation in collaborative filtering. The *k*-nearest neighbors of users are computed using the training set, and then the predictions are generated based on the testing set with equation (15). Figures1 to 4 show the performance of recommendations based on different similarity measures over the *MAE* and *RMSE* measures on the *Film Trust* and *Movie Lens* data sets, in which we vary the number of k nearest neighbors for each item from 30 to 300. In both figures 1 and 2, it can be observed that our proposed similarity based CF makes significantly less errors compared to the all other CFs which utilize state-of-the-art similarity measures. As a result, despite increasing MAE of the most CFs with increasing the number of nearest neighbors, the proposed *ICCF* based CF can achieve non-decreasing accuracy and lower MAE values. With k=300, the proposed *ICCF* based CF can generate the less mean absolute errors, MAE=0.77 in *FilmTrustand* MAE=0.21 in *MovieLens*, whereas the BCF_{cor} based CF which is the second best performing measure can generate MAE values close to 0.91 in FilmTrust and close to 0.34 in FilmTrust. The RMSE results for FilmTrust data set are shown in Fig. 3.It can be noted that the proposed CF_{ICCF} reduce error more than 9% compared to $CF_{BCF(corr)}$ and $CF_{BCF(med)}$ which have better performance among all the standard CFs. The *RMSE* results for MovieLens data set are shown in Fig. 4. It is shown that our proposed CF outperforms other CFs (with *RMSE*<0.36). Beside that, we can see that the proposed CF_{ICCF} improves accuracy with the increasing the number of nearest neighbors.

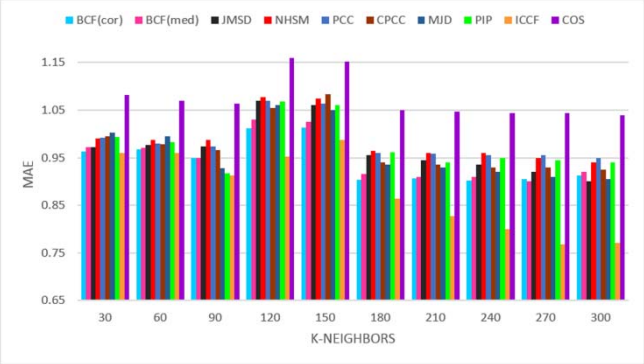

Fig. 1. The MAE analysis of different similarity measures on FilmTrust.

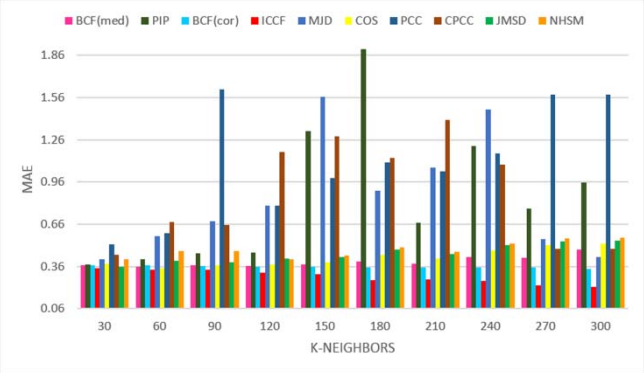

Fig. 2. The MAE analysis of different similarity measures on MovieLens.

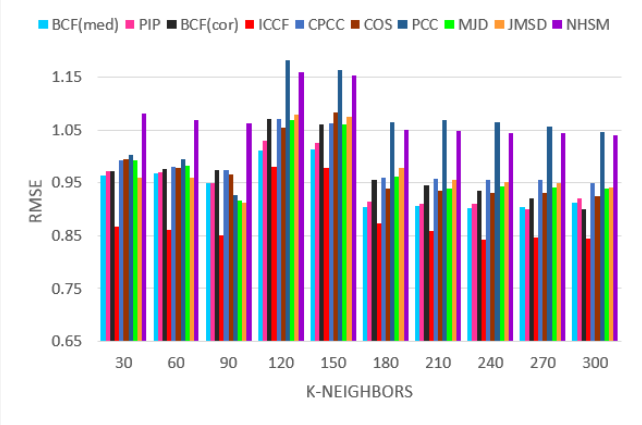

Fig. 3. The RMSE analysis of different similarity measures on FilmTrust.

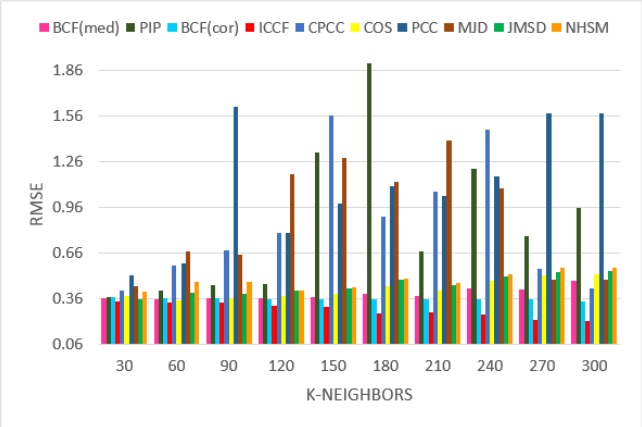

Fig. 4. The RMSE analysis of different similarity measures on MovieLens.

7. Conclusion

In this paper we focused on addressing the problems of sparsity and cold-start users associated with a recommender system. We proposed a new user similarity model to improve the neighborhood based collaborative filtering algorithm. We applied all-distance sketch node labels in item-item graph and also we took the proportion of common features between two users' rated items to compute closeness of the corresponding item sets. The experimental results on two benchmark datasets of *MovieLens* and *FilmTrust* with different scales and sparsity levels show that the proposed similarity measure is highly effective.

In this work, we have created the item-item graph to compute closeness similarity, this computation's query time will vary with the size of data. An important avenue for future work is to decrease similarity calculation's query time in very large data by applying an appropriate clustering method to create item sub-graphs, which is currently under development by the authors. Another important directions for future research is incorporating the impact of negative item ratings into similarity measurement.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey. Knowledge Based Systems. 2013. 26, 109–132.

[2] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art andpossible extensions. IEEE Trans. Knowl. Data Eng. 2005. 17(6), 734–749.

[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work. 1994.

[4] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. 2007. 58(7), 1019–1031.

[5] E. Cohen, H. Kaplan, Summarizing data using bottom-k sketches. In Proceedings of the ACM PODC'07 Conference. 2007.

[6] E. Cohen, All-distances sketches, revisited: Scalable estimation of the distance distribution and centralities in massive graphs. CoRR, vol. abs/1306-3284.2013a.

[7] E. Cohen, D. Delling, F. Fuchs, A. Goldberg, M. Goldszmidt, R. Werneck, Scalable similarity estimation in social networks: Closeness, node labels, and random edge lengths. In Proceedings of the first ACM conference on Online social networks. 2013. 131-142.

[8] G. Koutrica, B. Bercovitz, H. Garcia, FlexRecs: expressing and combiningflexible recommendations, in: Proceedings of the ACM SIGMOD International Conference on Management of Data. 2009. pp. 745–758.

[9] U. Shardanand, P. Maes, Social information filtering: algorithms for automating word of mouth. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 1994. 210–217.

[10] J. Bobadilla, F. Serradilla, J. Bernal, A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems.2010. 23, 520–528.

[11] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Information Science. 2008. 178(1), 37–51.

[12] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem. Knowledge Based Systems. 2012. 26, 225–238.

[13] H. Liu, Z. Hu, A. Mian, H. Tian, X. Zhu, A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems. 2014. 56, 156-166.

[14] B. K. Patra, R. Launonen, V. Ollikainen, S. Nandi, A new similarity measure using bhattacharyya coefficient for collaborative filtering in sparse data. Knowledge-Based Systems. 2015. 82, 163-177.

[15] I. Abraham, D. Delling, A.V. Goldberg, R.F. Werneck, (2012). Hierarchical Hub Labelings for Shortest Paths. in Proceedings of the 20th Annual European Symposium on Algorithms (ESA'12), Lecture Notes in Computer Science 7501, pp. 24–35. Springer, 2012.

[16] T. Akiba, Y. Iwata, Y. Yoshida, Fast exact shortest-path distance queries on large networks by pruned landmark labeling In Proceedings of the 2013 international conference on Management of data, 349-360. ACM.

[17] E. M. Maron, Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*. 1961.*8*(3), 404-417.

[18] A. H. Cheetham, J. E. Hazel, Binary (presence-absence) similarity coefficients. *Journal of Paleontology*. 1969. 1130-1136.

[19] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate thenew user cold-starting problem, Inform. Sci. 2008. 178 (1) 37–51.

[20] L. Christophe, et al., Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of Experimental Social Psychology. 2013. 49(4), 764-766.

[21] L.A. Adamic, E. Adar, How to search a social network. Social Networks. 2005. 27(3), 187-203.

[22] E. Cohen, (2014). Variance Competitiveness for Monotone Estimation: Tightening the Bounds. arXiv preprint arXiv, 1406.6490.

[23] E. Cohen, (2013b). Estimation for monotone sampling: Competitiveness and customization. PODS. ACM.

[24] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. T. Riedl., Evaluating collaborative filtering recommender systems.*ACM Transactions on Information Systems* (*TOIS*) 22.1. 2004. 5-53.