Knowledge-Based Systems xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data

Bidyut Kr. Patra [a,c,*], Raimo Launonen [a], Ville Ollikainen [a], Sukumar Nandi [b]

[a] VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, Finland
[b] Indian Institute of Technology Guwahati, Guwahati, Assam 781 039, India
[c] National Institute of Technology Rourkela, Rourkela, Odisha 769 008, India

ARTICLE INFO

Article history:
Received 11 March 2014
Received in revised form 5 February 2015
Accepted 1 March 2015
Available online xxxx

Keywords:
Collaborative filtering
Neighborhood based CF
Similarity measure
Bhattacharyya coefficient
Sparsity problem

ABSTRACT

Collaborative filtering (CF) is the most successful approach for personalized product or service recommendations. Neighborhood based collaborative filtering is an important class of CF, which is simple, intuitive and efficient product recommender system widely used in commercial domain. Typically, neighborhood-based CF uses a similarity measure for finding similar users to an active user or similar products on which she rated. Traditional similarity measures utilize ratings of only co-rated items while computing similarity between a pair of users. Therefore, these measures are not suitable in a sparse data. In this paper, we propose a similarity measure for neighborhood based CF, which uses all ratings made by a pair of users. Proposed measure finds importance of each pair of rated items by exploiting Bhattacharyya similarity. To show effectiveness of the measure, we compared performances of neighborhood based CFs using state-of-the-art similarity measures with the proposed measured based CF. Recommendation results on a set of real data show that proposed measure based CF outperforms existing measures based CFs in various evaluation metrics.

1. Introduction

Recommender System (RS) techniques have been successfully used to help people cope with information overload problem and they have been established as an integral part of e-business domain over last decades. The primary task of a recommender system is to provide personalized suggestions for products or items to individual user filtering through large product or item space. Many recommender system algorithms have been developed in various applications such as e-commerce, digital library, electronic media, and on-line advertising [1–4]. These algorithms can be categorized into two major classes, *viz. content based filtering* and *collaborative filtering*, and combination of these can also be found in the literature to address their disadvantages.

In content-based filtering [5,6], an item is recommended to an active user by analyzing profile of the user, profile of the item and profiles of items she preferred in past. However, analyzing profiles is often hard in many applications such as multimedia data [7].

Collaborative filtering (CF) is the most successful and widely used recommendation system [7–9]. In CF, item recommendations to a user are performed by analyzing rating information of the other users or other items in the system. The main advantages of collaborative filtering are that it is domain independent and more accurate than content based filtering. There are two main approaches for recommending items in CF category, *viz. neighborhood based CF* and *model based CF*.

Neighborhood based CF rely on a simple intuition that an item might be interesting to an active user if the item is appreciated by a set of similar users (neighbors) or she has appreciated similar items in the system.

Model-based CF algorithms learn a model from the training data using machine learning and other techniques [10,11,9]. Subsequently, the model is used for predictions. Main advantage of the model-based approach is that it does not need to access whole rating data once model is built. Few model based approaches provide more accurate results than neighborhood based CF [12,13]. However, most of the electronic retailers such as Amazon, Netflix deployed neighborhood based recommender systems to help out their customers. This is due to the fact that

* Corresponding author at: VTT Technical Research Centre of Finland, P.O. Box 1000, FI-02044 VTT, Finland.
E-mail addresses: ext-bidyut.patra@vtt.fi (B.Kr. Patra), raimo.launonen@vtt.fi (R. Launonen), ville.ollikainen@vtt.fi (V. Ollikainen), sukumar@iitg.ernet.in (S. Nandi).

http://dx.doi.org/10.1016/j.knosys.2015.03.001
0950-7051/

neighborhood based approach is simple, intuitive and it does not have learning phase so it can provide immediate response to new user after receiving upon her feedback [14]. One more advantage of neighborhood based approach is that it works with a single parameter (K-number of neighborhood) unlike model based approach which needs many parameters (learning parameter η, regularization parameters ν, etc.).

Generally, neighborhood based CF uses a similarity measure for finding neighbors of an active user or finding similar items to the candidate item. Traditional similarity measures such as pearson correlation coefficient, cosine similarity and their variants are frequently used for computing similarity between a pair of users or between a pair of items [15]. In these measures, similarity between a pair of users is computed based on the ratings made by both users on the common items (co-rated item). Likewise, item similarity is computed using the ratings provided by users who rated both the items. However, correlation based measures perform poorly if there are no sufficient numbers of co-rated items in a given rating data. For example, two items can be similar if there is no single user who rates both the items. Likewise, two users can be similar if they rated different items. Therefore, correlation based measure and its variants are not suitable in a sparse data in which number of ratings by individual user is less and number of co-rated items is few or none. Yildirim and Krishnamoorthy observed that correlation based similarity measure is not suitable in sparse data [16].

In this paper, we propose a novel approach for finding similarity between a pair of users in sparse data. Proposed measure gives importance to each rating made by the pair of users. The Bhattacharyya measure [17] (popular in signal and image processing domains) is utilized for finding the relevance between a pair of rated items. The proposed similarity measure is termed as Bhattacharyya Coefficient in CF (BCF). The BCF measured based CF is tested on real rating datasets. Our contributions in this paper are summarized as follow.

- A novel similarity measure for user based collaborative filter is proposed. Unlike existing measures, proposed measure uses all ratings made by a pair of users.
- The Bhattacharyya measure is utilized in BCF. The Bhattacharyya measure plays an important role for finding relevance between each pair of rated items. We do not assume rating distribution of item. The BCF combines rating correlation with relevance of each pair of rated items. The BCF can compute similarity between two users in the absence of co-rated items.
- To show effectiveness of the proposed measure we implemented neighborhood based CF using correlation based measures and using its variants. Recommendation results on three popular datasets are tested using various performances metrics such as *Mean Absolute Error (MAE)*, *Root Mean Squared Error (RMSE)*, F1 measure. The CF using proposed BCF measure outperforms CFs using state-of-the-art measures.

The rest of the paper is structured as follows. In Section 2, we discuss necessary background and related work. In Section 3, we present our similarity measure. Experimental results of BCF based CF are provided in Section 4. We conclude the paper in Section 5 with possible future research directions.

2. Background and related work

In this section, we discuss working principle of neighborhood based approach in detail and different similarity measures introduced in-order to increase performance of recommendation systems.

2.1. Neighborhood-based approach

The neighborhood or memory based approach is introduced in the GroupLens Usenet article recommender [18] and has gained popularity due to its wide application in commercial domain [3,14,19]. This approach uses the entire rating dataset to generate a prediction for an item (product) or a list of recommended items for an active user. Let $\mathcal{R} = [r_{ui}]^{M\times N}$ be a given rating matrix (dataset) in a CF based recommender system, where each entry r_{ui} represents a rating value made by uth user U_u on ith item I_i. Generally, rating values are integers within a range (say, 1–5 in MovieLens dataset). An entry $r_{ui} = 0$ indicates user U_u has not rated the item I_i. The prediction task of neighborhood-based CF algorithm is to predict rating of the ith item either using the neighborhood information of uth user (*user-based method*) or using neighborhood information of ith item (*item-based method*).

The user-based methods predict based on ratings of ith item made by the neighbors of the uth user [15,20]. For this purpose, this method computes similarity between the active user (here, U_u user) and $U_p,\ p = 1\ldots M,\ p \neq u$. Then, it selects K closest users to form the neighborhood of the active user. Finally, it predicts a rating $\hat{r}_{ui}$ of the ith item using the following Eq. (1).

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{k=1}^{K} s(U_u, U_k) \ * \ (r_{ki} - \bar{r}_{k_u})}{\sum_{k=1}^{K} |s(U_u, U_k)|} \tag{1}$$

where $\bar{r}_u$ is the average of the ratings made by the user U_u; $s(U_u, U_k)$ denotes the similarity value between user U_u and its kth neighbor; $\bar{r}_{k_u}$ is the average of the ratings made by kth neighbor of the user U_u, and r_{ki} is the rating made by kth neighbor on ith item.

Item-based collaborative filtering has been deployed by world's largest online retailer Amazon Inc. [3]. The item-based method is introduced in [21,22]. It computes similarity between target item I_i and all other items $I_j,\ j = 1\ldots N,\ i \neq j$ to find K most similar items. Finally, unknown rating $\hat{r}_{ui}$ is predicted using the ratings on these K items made by the active user U_u (Eq. (2)).

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{k=1}^{K} s(I_i, I_k) \ * \ (r_{uk} - \bar{r}_k)}{\sum_{k=1}^{K} |s(I_i, I_k)|} \tag{2}$$

where $\bar{r}_i$ is the average of the ratings made by all users on items I_i; $s(I_i, I_k)$ denotes the similarity between the target item I_i and the kth similar item, and r_{uk} is the rating made by the active user on the kth similar item of I_i.

Similarity computation is a vital step in the neighborhood based collaborative filtering. Many similarity measures have been introduced in various domains such as machine learning, information retrieval, and statistics. Researchers and practitioners in recommender system community used them directly or invented new similarity measure to suit the purpose. We discuss them briefly next.

2.2. Similarity measure in CF

Traditional measures such as pearson correlation coefficient (PC), cosine similarity are frequently used in recommendation systems. A list of popular similarity measures used in neighborhood based CF is given in Table 1. The cosine similarity is very popular measure in information retrieval domain [23]. To compute similarity between two users U and V, they are considered as the two rating vectors of N dimensions, *i.e.*, $U, V \in \mathbb{N}_0^N$, where $\mathbb{N}_0$ is the set of natural numbers including 0. Then, similarity value between two users is the cosine of the angle between U and V (Table 1). Cosine similarity is popular in item based CF. However, cosine similarity does not consider the different rating scales (ranges) provided by the individual user while computing similarity between a pair of items. Adjusted cosine similarity measure

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

Table 1
Similarity measures frequently used in neighborhood based CF.

Measure	Definition/formula	Major drawbacks
Cosine [23]	$s(U,V)=\frac{\sum_{I\in I'}(r_{UI})(r_{VI})}{\sqrt{\sum_{I\in I'}r_{UI}^2}\sqrt{\sum_{I\in I'}r_{VI}^2}}$ where r_{UI} is the rating made by user U on item I and I' is the set of co-rated items	It suffers from *few co-rated item* problem It outputs high similarity in spite of significant difference in ratings
Adjusted Cosine [21]	$s(I,J)=\frac{\sum_{U\in U'}(r_{UI}-\bar{r}_I)(r_{UJ}-\bar{r}_J)}{\sqrt{(r_{UI}-\bar{r}_I)^2}\sqrt{(r_{UJ}-\bar{r}_J)^2}}$ where r_{UI} is the rating made by user U on item I, $\bar{r}_I$ is the average rating of item I and U' is the set of users rated both items	It cannot compute similarity if carnality of U' is small. It shows low (high) similarity regardless of similar (significant difference in) the ratings
Pearson Correlation (PC) [20]	$s(U,V)=\frac{\sum_{I\in I'}(r_{UI}-\bar{r}_U)(r_{VI}-\bar{r}_V)}{\sqrt{\sum_{I\in I'}(r_{UI}-\bar{r}_U)^2}\sqrt{\sum_{I\in I'}(r_{VI}-\bar{r}_V)^2}}$ where I' is the set of co-rated items	It suffers from *few co-rated item* problem It shows low (high) similarity regardless of similar (difference) in the ratings
Constrained PC (CPC) [24]	$s(U,V)=\frac{\sum_{I\in I'}(r_{UI}-r_{med})(r_{VI}-r_{med})}{\sqrt{\sum_{I\in I'}(r_{UI}-r_{med})^2}\sqrt{\sum_{I\in I'}(r_{VI}-r_{med})^2}}$ where r_{med} is the median value in rating scale and I' denotes set of co-rated items	It addresses the problem of PC but suffers from *few co-rated item* problem
Mean squared difference (MSD) [24]	$s(U,V)=1-\frac{\sum_{I\in I'}(r_{UI}-r_{VI})^2}{\lvert I'\rvert}$ where I' is the set of co-rated items and r_{UI} is the rating of user U on item I	It ignores proportion of common ratings
Jaccard	$s(U,V)=\frac{\lvert I_U\cap I_V\rvert}{\lvert I_U\cup I_V\rvert}$ where I_U is the set of items rated by user U	It does not take absolute value (rating) into account
JMSD [25]	$s(U,V)=s_{MSD}(U,V)\times s_{Jac}(U,V)$ where $s_{MSD}(\cdot)$ and $s_{Jac}(\cdot)$ are similarity in MSD and Jaccard measures, respectively	It addresses the MSD and Jaccard partially It suffers from *local information* and *utilization of rating* problems

addresses this drawback by subtracting the corresponding user average from the rating of the item [21]. It computes linear correlation between ratings of the two items (Table 1).

Pearson correlation coefficient (PC) is very popular measure in user-based collaborative filtering. The PC measures how two users (items) are linearly related to each other. Having identified co-rated items between users U and V, PC computes correlation between them using definition given in Table 1. The value of PC ranges in $[-1,+1]$. The value $+1$ indicates highly correlated and -1 indicates negatively co-related to each other. Likewise, similarity between two items I and J can also be computed using PC. Constrained Pearson correlation coefficient (CPC) is a variant of PC in which an absolute reference (median in the rating scale) is used instead of corresponding user's rating average (Table 1). The mean-squared difference (MSD) is proposed in [24] and it is computed using the formula given in Table 1. But it has not been received attention of researchers subsequently. PC and its variants suffer from the following drawbacks.

- *Few co-rated items*: These measures cannot compute similarity between users if there are few co-rated items. Likewise, item similarity suffers from the same drawback. Two items can also be very similar if there is no single user who rates both the items. Let $I=(1,0,2,0,1,0,2,0,3,0)^T$ and $J=(0,1,0,2,0,1,0,2,0,3)^T$ be the rating vectors corresponding to a pair of items on which no user rates both the items. It can be noted that the rating patterns are identical, which cannot be captured by these measure. Therefore, these are not suitable in a scenario in which number of ratings of individual user is few as these reduces the chances of having many co-rated items.
- *Number of co-rated item only one*: If number of co-rated item between two users is exactly one, then similarity between them cannot be computed using PC or cosine measure [26]. The PC value between them is either $+1$ or -1 and cosine value is 1, regardless of their ratings on the item.
- *Local information*: These measures use only local information of the ratings while computing similarity between a pair of users. They do not use global information of the items on which a pair of users rated.
- *Utilization of ratings:* These measures do not utilize all ratings provided by the pair of users.

Therefore, these measure cannot be used for neighborhood based CF in sparse rating data. The Jaccard similarity measure uses all ratings information provided by a pair of users (Table 1). However, it uses numerical (absolute) values of the ratings and suffers from *few co-rated items* problem (number of co-rated items is directly proportional to the similarity value).

To improve the recommendation accuracy of neighborhood based CF, many researchers introduced different similarity measure for solving data sparsity and cold start problem [8] in last few years. Luo et al. [27] address similarity problem in sparse data by introducing two types of similarity, namely local user similarity and global user similarity. For computing local similarity between a pair of users, they use each user's surprisal vector, which is derived from rating distribution of each rated item and rating made on the item. Final prediction is the linear combination of predictions obtained from local neighbors and global neighbors. This work is extended in [28]. Various sparsity measures are introduced and used them as weights in the linear combination. Main drawback of this approach is that rating distribution of each item is assumed to be Laplacian.

PIP is the most popular (cited) measure after traditional similarity measures in RS. The PIP measure captures three important aspects (factors) namely, *proximity*, *impact* and *popularity* between a pair of ratings on the same item [26]. The proximity factor is the simple arithmetic difference between two ratings on an item with an option of imposing penalty if they disagree in ratings. The agreement (disagreement) is decided with respect to an absolute reference, *i.e.*, median of the rating scale. The impact factor shows

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

how strongly an item is preferred or disliked by users. It imposes penalty if ratings are not in the same side of the median. Popularity factor gives important to a rating which is far away from the item's average rating. This factor captures global information of the concerned item. The PIP computes these three factors between each pair of co-rated items. PIP based CF outperforms correlation based CF in providing recommendations to the new users.

Kim et al. [29] proposed a similarity measure to address the cold-start problem. It builds a model which first predicts rating and computes prediction error on known ratings for each user. From this error information the final model is built. However, approach uses traditional similarity measures (PC, cosine) for initial predictions.

Bobadilla et al. proposed a number of similarity measures to address drawbacks of the traditional measures in [25,30,31]. Bobadilla et al. [25] proposed to combine Jaccard and Mean-squared-difference(MSD) (called JMSD) to complement each other. They also proposed singularity based measure, which combines percentage of relevant, non relevant ratings with the MSD of the co-rated items [30]. Bobadilla et al. proposed to use numerical ratings (information) as well as the distributions (variation) of ratings made by a pair of users while computing similarity between them in [31]. To capture the numerical information, authors compute numbers of co-rated items with exactly same rating, number of co-rated items with different rating scales (say, 1, 3, 4) and MSD on ratings on co-rated items. The Jaccard measure is used to capture the variation of ratings provided by the pair of users. These measures are termed as basic measures. Finally, these basic measures are combined to form a similarity measures which is termed as Mean-Jaccard-Difference (MJD). Neural learning technique is used to compute weight for each basic measure. They showed that the MJD based CF starts outperforming PIP based CF in MAE (*Mean Absolute Error*) accuracy measure after computing a large number of neighbors (K). However, these all three measures suffer from *few co-rated items* problem. Therefore, these could not be used in sparse rating data.

Choi and Suh [32] introduced a PC based measure and they argued that neighborhood of an active user is dynamic and it depends on the target item. They computed similarity between a user V with an active user U for a target item I using Eq. (3) given below.

$$s^{I}(U,V) = \frac{\sum_{J\in I'} s(I,J)^2 (r_{UJ} - \bar{r}_U)(r_{VJ} - \bar{r}_V)}{\sqrt{\sum_{J\in I'} \{s(I,J)(r_{UJ} - \bar{r}_U)\}^2} \sqrt{\sum_{J\in I'} \{s(I,J)(r_{VJ} - \bar{r}_V)\}^2}} \quad (3)$$

where $s(I,J)$ is the similarity between the target item I and a co-rated item J, I' is the set of co-rated items; r_{UJ} and $\bar{r}_U$ are the rating of user U on the item J and average rating of user U, respectively. The PC measure is used to compute item similarity. This measure also uses only co-rated items.

Haifeng Liu et al. introduced a new similarity measure called NHSM (new heuristic similarity model), which addresses the drawbacks of PIP based measure recently [33]. They put an argument that PIP based measure unnecessarily penalizes more than once while computing *proximity* and *impact* factors. They adopted a non-linear function for computing three factors, namely, *proximity, significance, singularity* in the same line of PIP based measure. Finally, these factors are combined with modified Jaccard similarity measure [33]. However, these factors are computed only on co-rated items. Ratings on non-corated items are neglected. To utilize ratings of non-corated items, popular Bhattacharyya measure is exploited first in [34]. Patra et al. [34] proposed a generalized formula for similarity measure in which existing measures can be incorporated to address user cold-start problem. In this approach, ratings on a pair of non-corated items are taken into account if similarity between the non-corated items is maximum. Otherwise, ratings are ignored. Main drawback of this approach is that it cannot utilize ratings of all non co-rated items. This approach cannot be used in finding similarity between a pair of users if they rate on few or no similar items (maximum similarity). Zhen et al. [35–37] proposed CF based recommender systems for collaborative team environments. As traditional measures cannot be used in this scenario, they introduce a similarity measure which captures context information of the collaborative team [35]. The four dimensional work-flow space model is exploited for finding similarity between team members. Zhen et al. [37] also proposed a measure for computing similarity between peers for knowledge recommender system in P2P environments. The profile information of each peer consists of numeric, binary and nominal attributes. These three types of attributes are treated separately. These measures are highly domain specific.

3. Proposed similarity measure for CF

The critical step of neighborhood-based CF approach is to find neighbors of an active user using suitable similarity measure. As we discussed in previous section, traditional similarity measures cannot be used in sparse rating dataset. In this section, we propose our similarity measure, which is suitable in sparse dataset. The motivations of our proposed measure are as follow.

- In a sparse data, number of rating by an individual user is few in general and co-rated items are found to be rare. Proposed measure works with few or no co-rated items between a pair of users.
- The similarity measure use local as well as global ratings information. The local information computed using correlation of the users' ratings. The global information is extracted in terms of similarity between a pair of items. The Bhattacharyya similarity measure is exploited for this purpose.
- The measures discussed in the previous section could not use all ratings (absolute value) information efficiently. The Jaccard and JMSD are successful to some extent. Our measure utilizes all ratings (absolute value) provided by a pair of users.

The proposed measure uses Bhattacharyya measure, which is introduced to compute distance (divergence) between two probability distributions [17]. First, we discuss Bhattacharyya measure and subsequently show how this measure can intelligently be used to remove data sparsity problem in recommender system.

3.1. Bhattacharyya measure

The Bhattacharyya measure has been widely used in signal processing, image processing and pattern recognition research community [38–42]. It measures similarity between two probability distributions. Let $p_1(x)$ and $p_2(x)$ be two density distributions over continuous domain. Then, Bhattacharyya Coefficient (BC) (similarity) between these densities is defined [41] as

$$BC(p_1,p_2) = \int \sqrt{p_1(x)\, p_2(x)}\, dx \quad (4)$$

The BC is defined over a discrete domain X as follows Eq. (5).

$$BC(p_1,p_2) = \sum_{x\in X} \sqrt{p_1(x)\, p_2(x)} \quad (5)$$

Densities of $p_1(x)$ and $p_2(x)$ are estimated from the given rating data. Histogram formulation can be used to estimate these densities as used in [39,40]. Let $\hat{p}_i$ and $\hat{p}_j$ be the estimated discrete densities of

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

the two items i and j obtained from rating data. Then, BC similarity between item i and item j is computed as

$$BC(i,j) = BC(\hat{p}_i, \hat{p}_j) = \sum_{h=1}^{m} \sqrt{(\hat{p}_{ih})\,(\hat{p}_{jh})} \tag{6}$$

where m is the number of bins and $\hat{p}_{ih} = \frac{\#h}{\#i}$, where $\#i$ is the number of users rated the item i, $\#h$ is the number of users rated the item i with rating value 'h', $\sum_{h=1}^{m} \hat{p}_{ih} = \sum_{h=1}^{m} \hat{p}_{jh} = 1$.

This can be illustrated with an example. Let $I = (1,0,2,0,1,0,2,0,3,0)^T$ and $J = (0,1,0,2,0,1,0,2,0,3)^T$ be the rating vectors of items I and J, respectively. The ratings lie in $\{1,2,3\}$. Then, BC coefficient between I and J:

$$BC(I,J) = \sum_{h=1}^{3} \sqrt{\hat{I}_h \hat{J}_h} = \sqrt{\left(\frac{2}{5}\right) * \left(\frac{2}{5}\right)} + \sqrt{\left(\frac{2}{5}\right) * \left(\frac{2}{5}\right)} + \sqrt{\left(\frac{1}{5}\right) * \left(\frac{1}{5}\right)} = 1.$$

It may be noted that there is no single user who has rated both the items. Existing measures could not compute item similarity in this scenario.

3.2. BCF: A similarity measure for neighborhood based CF in sparse data

The proposed similarity measure is termed as Bhattacharyya Coefficient in CF (BCF). The BCF utilizes numerical values of all ratings made by a pair of users, not only the ratings made on common items. The BCF combines local and global similarity to obtain final similarity value. Let I_U and I_V be the two sets of items on which users U and V rated, respectively. It may happen that there is no co-rated item ($I_U \cap I_V = \emptyset$) between U and V. The similarity between the users U and V in BCF measure is the function of BC coefficient between a pair of rated items and local similarity between the ratings on the pair of items (Eq. (7)).

$$s(U,V) = \sum_{i \in I_U} \sum_{j \in I_V} BC(i,j)\, loc(r_{Ui}, r_{Vj}) \tag{7}$$

The $BC(i,j)$ provides global ratings information of the pair of items i, j and $loc(\cdot)$ finds local similarity between two ratings. As we observed in the previous subsection that BC can compute similarity between i and j even though there is no single user who rates both the items. If two items are similar in global perspective, then $BC(i,j)$ enhances local similarity between ratings of the corresponding users on the items i and j (Eq. (7)). On the other hand, if items i and j are dissimilar to each other, then $BC(i,j)$ decreases importance of local similarity between ratings on the pair of items.

The local similarity plays an important role and it provides local user information. The local similarity must provide positive as well as negative correlation between user ratings. The local similarity between a pair of ratings $loc(r_{Ui}, r_{Vj})$ can be evaluated using two functions. The first function evaluates correlation between these two ratings using Eq. (8).

$$loc_{cor}(r_{Ui}, r_{Vj}) = \frac{(r_{Ui} - \bar{r}_U)(r_{Vj} - \bar{r}_V)}{\sigma_U \sigma_V} \tag{8}$$

where $\bar{r}_U$ is the average of the ratings made by the user U, r_{Ui} is the rating made by the user U on item i, σ_U is the standard deviation of the ratings made by the user U.

The function $loc_{cor}(\cdot)$ considers user's average rating as the reference scale. However, median of the ratings scale can also be considered for this purpose. Therefore, we propose another function $loc_{med}(r_{Ui}, r_{Vi})$ for computing local similarity between a pair of ratings r_{Ui} and r_{Vi} (Eq. (9)).

$$loc_{med}(r_{Ui}, r_{Vi}) = \frac{(r_{Ui} - r_{med})(r_{Vi} - r_{med})}{\sqrt{\sum_{k \in I_U} (r_{Uk} - r_{med})^2} \sqrt{\sum_{k \in I_V} (r_{Vk} - r_{med})^2}} \tag{9}$$

where r_{med} is the median of the rating scale, I_U is the set of items rated by the user U and r_{Uk} is the rating made by the user U on item k.

The BCF provides maximum importance to local similarity if ratings are made on a same item as the item-similarity on same item is 1 ($BC(i,i) = 1$). It does not give any importance to local similarity if a pair of users made ratings on totally dissimilar items ($BC(i,j) = 0$). To provide more importance to the number of common items, Jaccard similarity measure $Jacc(U,V)$ between user U and V is added with $s(U,V)$. Therefore, Eq. (7) is modified as

$$s(U,V) = Jacc(U,V) + \sum_{i \in I_U} \sum_{j \in I_V} BC(i,j)\, loc(r_{Ui}, r_{Vj}) \tag{10}$$

3.3. Discussion on proposed similarity measure

Here, we discuss some important properties of the proposed similarity measure.

- *Few or no co-rated items:* All existing similarity measures fail to compute neighborhood of an active user if the active user does not rate a minimum number of items on which a significant number of users rated. However, our proposed BCF can compute neighborhood of the active user in this scenario as BCF does not depend on number of co-rated items.
- *Local and global information:* The BCF extracts global information from the sparse rating data which is useful for computing similarity between a pair of users. This enhances the possibility of finding similar users to an active user in the sparse data. The $BC(\cdot)$ is exploited for finding global similarity between a pair of items (Eq. (6)). Proposed BCF also computes local information (similarity) using either function $loc_{cor}(\cdot)$ or $loc_{med}(\cdot)$.
- *Utilization of all absolute ratings:* Jaccard and its variant JMSD use all rating information in limited sense (could not use numerical value of the ratings). On the other hand, BCF computes each pair of ratings' local user similarity and similarity between the rated items, which comprehensively use numerical values of the ratings. (Eq. (10)).

4. Experimental evaluation

To evaluate performance of proposed similarity based CF, we implemented user-based collaborative filterings using different existing similarity measures. We used traditional similarity measures such as PC and CPC, PIP measure (designed for cold-start problem) [26], JMSD (combination of Jaccard and MSD) [25], MJD (combination of several basic similarity measures) [31] and recently proposed NHSM (improved PIP) [33]. It may be noted that all these measures are discussed in Section 2.2. For the shake of readability, we use notation to denote a collaborative filtering using a specific similarity measure, *i.e.* CF_{PC} denotes collaborative filtering using PC measure and other notations are given in Table 2.

4.1. Data preparation

We used three different real datasets, namely MovieLens,[1] Netflix[2] and Yahoo Music[3] in experiments. Brief description of these three datasets is given in Table 3. To show effectiveness of our BCF

[1] http://www.grouplens.org.
[2] http://www.netflixprize.com.
[3] http://research.yahoo.com/academic_relations.

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

Table 2
Description of notations used for different CFs.

Measured used	Corresponding CF
BCF_{cor}	$CF_{BCF(cor)}$
BCF_{med}	$CF_{BCF(med)}$
MJD	CF_{MJD}
JMSD	CF_{JMSD}
PIP	CF_{PIP}
NHSM	CF_{NHSM}
PC	CF_{PC}
CPC	CF_{CPC}

Table 3
Description of the datasets used in the experiments.

Dataset	Purpose	#User	#Item	#Rating	κ	Rating domain
MovieLens	Movie	6040	3706	1 M	4.46	$\{1,2,3,4,5\}$
Netflix	Movie	480,189	17,770	100 M	1.17	$\{1,2,3,4,5\}$
Yahoo	Music	15,400	1000	0.3 M	1.94	$\{1,2,3,4,5\}$

Table 4
Statistics of sparse subsets.

Dataset (original)	Data-subset	#User (M)	#Item (N)	#Rating (R)	κ $\left(\frac{R\times100}{M\times N}\right)$	$\frac{R}{M}$	$\frac{R}{N}$
Movie-Lens	ML_1	6040	3706	40,957	0.18	6.8	11.1
	ML_2	1000	2994	6000	0.20	6.0	2.0
Netflix	Net_1	8141	9318	196,656	0.25	24.2	21.1
	Net_2	8141	9318	72,184	0.10	8.8	7.4
Yahoo	YM	15,400	1000	49,892	0.32	3.2	49.8

measure in sparse data, we obtained total five subsets in various sparsity levels removing ratings randomly from these datasets. The sparsity level is parameterized by the *density index* (κ), which is the percentage of all possible ratings available in a dataset. The characteristics of all these subsets is summarized in Table 4.

4.2. Evaluation metrics

Research communities in RS have used several types of evaluation metrics to compare quality of recommender systems [43]. These can be broadly classified into two categories: *Predictive Accuracy* and *Classification Accuracy* metrics.

Predictive Accuracy: It evaluates quantitative accuracy of an prediction value on a target item. Two popular metrics *Mean Absolute Error (MAE)* and *Root Mean Squared Error (RMSE)* are used for this purpose. The *MAE* (Eq. (11)) is the average absolute errors over all predictions and a smaller value indicates a better accuracy.

$$MAE = \frac{\sum_{i=1}^{MAX}|r_i - \hat{r}_i|}{MAX} \tag{11}$$

where r_i and $\hat{r}_i$ are the actual rating and predicted rating of an active user on an item by a CF algorithm, respectively; *MAX* is the number of times the predictions are performed by a CF algorithm. Similarly, RMSE is calculated using the following Eqs. (12).

$$RMSE = \sqrt{\frac{1}{MAX}\sum_{i=1}^{MAX}(r_i - \hat{r}_i)^2} \tag{12}$$

Classification Accuracy: It measures qualitative performance of a recommender system. Many recommender systems provide a list of items L_r to an active user instead of predicting ratings. There are two popular metrics to evaluate quality of a RS in this scenario: (i) *Precision*, which is the fraction of items in L_r that are relevant and (ii) *Recall*, which is the fraction of total relevant items that are in the recommended list L_r. A list of relevant items L_{rev} to a user is the set of items on which she made high ratings (i.e., $\geqslant 4$ in MovieLens dataset) in the test set. Therefore, Precision and Recall can be written as follow.

$$Precision = \frac{|L_r \cap L_{rev}|}{|L_r|} \text{ and } Recall = \frac{|L_r \cap L_{rev}|}{|L_{rev}|}$$

However, there is always a trade-off between these two measures. For instance, increasing the number of items in L_r increases Recall but decreases Precision. Therefore, we use a measure which combines both called F1 measure (Eq. (13)) in our experiments.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{13}$$

4.3. Experimental results and analysis

We analyzed characteristics of each subset. In this study, we consider each user as an active user and find average number of users who rate on same items as the active user rates. Fig. 1 shows that ML_1 subset has an average of less than 275 users (out of 6039) who share only one (1) co-rated item and it has an average of only 6 users who share two (2) co-rated items with each (active) user. The ML_2 has an average of less than 20 users who share only one co-rated item. In Net_1 subset, 570, 57 and 13 users share one, two and three co-rated items, respectively (Fig. 1). The Net_2 and *YM* have 247 and 511 users who share only one co-rated item, respectively.

To evaluate performance of our proposed BCF measure, we selected a fixed number (say, 5) of relevant items randomly as the target items for each user and compute *MAE*, *RMSE*, number of *successful prediction* and number of *perfect prediction* for these selected target items. Many neighborhood based CF cannot make a valid prediction for a target item as no neighbor rates the item. Therefore, number of valid predictions is also an important metric. The number of *successful prediction* is the number of valid prediction made by a CF. The number of *perfect prediction* is the number of times a CF correctly predicts the actual ratings. We also computed F1 measure for each CF mentioned earlier.

Fig. 2 shows MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) obtained after getting executed different CFs (Table 2) on ML_1 subset. It can be noted that our proposed similarity based both CFs make significantly less errors compared to the all other CFs which use state-of-the-art similarity measures (MJD, NHSM, JMSD, PIP). The $CF_{BCF(cor)}$ makes least error in both the metrics (Fig. 2). The CFs using state-of-the art similarity measures could not utilize all ratings information while computing neighborhood of an active user. These measures use only one rating (one co-rated item) information to include a user into the active user's neighborhood (Fig. 1). Therefore, collaborative filtering based on these measures make errors ($MAE \geqslant 0.822, RMSE \geqslant 1.07$) in predictions over a large number of nearest neighbors ($K = 40$ to 300) (Fig. 2). The CF_{MJD} is found to be the closest competitor in terms of both predictive accuracy metrics among the non traditional measures. However, $CF_{BCF(cor)}$ is the superior (*i.e.*, $MAE < 0.73$ and $RMSE < 1.00$). The $CF_{BCF(cor)}$ reduces *MAE* more than 10% compared to the CF_{MJD}. Similar trend is observed in *RMSE* metric.

The state-of-the-art neighborhood based CFs cannot make many *perfect predictions* compared to the our proposed *BCF*(*cor*) based collaborative filtering (Fig. 3). The proposed $CF_{BCF(cor)}$ makes consistently highest number of perfect predictions among all collaborative filterings discussed here over a large number of nearest neighbors. In Fig. 3, it can be observed that $CF_{BCF(cor)}$ predicts as

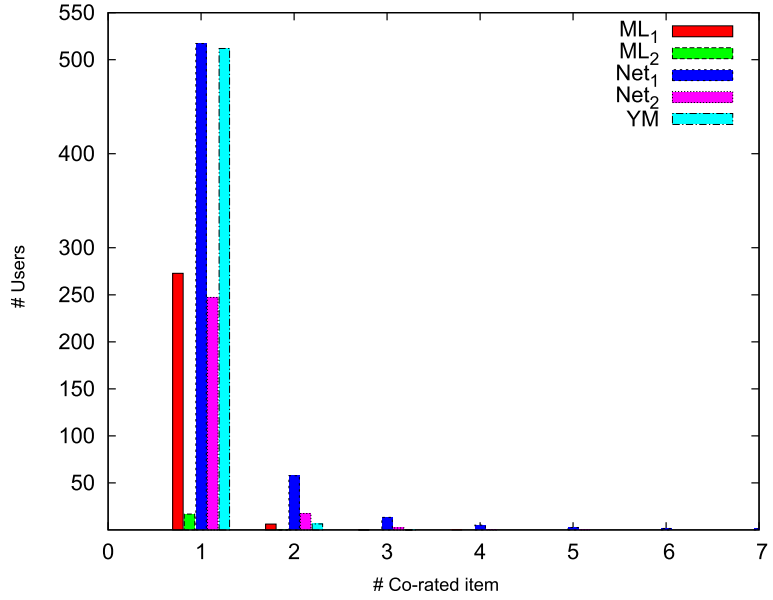

Fig. 1. Characteristics of data subsets.

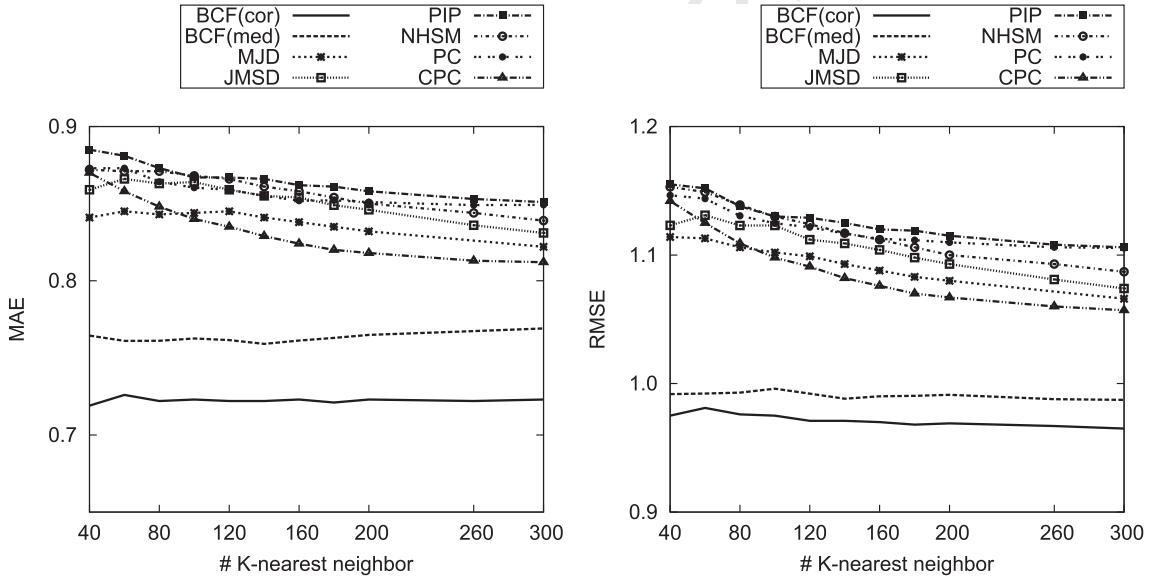

Fig. 2. MAE, RMSE vs K-nearest neighbors on ML_1 subset ($\kappa = 0.18$).

many as 2392 perfect predictions out of 21,646 successful predictions with $K = 200$. However, the closest competitor CF_{MJD} makes as many as 1267 perfect predictions (half in number compared to $CF_{BCF(cor)}$) with $K = 180$ out of 18,718 successful predictions (Fig. 3). Proposed $CF_{BCF(med)}$ makes least number of perfect predictions. However, it makes significant number of valid predictions. Proposed $CF_{BCF(cor)}$ makes highest number of successful predictions ($24,162$ with $K = 300$). This shows that BCF based measures are more accurate to draw useful neighborhood compared to the existing measures.

We computed F1 measure for each CF to show effectiveness of proposed similarity measures in the sparse ML_1 subset. The F1 values of each CF are plotted over different values of nearest neighbors and it is shown in Fig. 4. The proposed $CF_{BCF(cor)}$ and $CF_{BCF(med)}$ perform substantially well over all other neighborhood based collaborative filterings. The BCF based CFs can achieve F1 measure close to 0.65, whereas closest competitor CF_{MJD} can attain F1 measure close to 0.58 with $K = 300$., i.e., $CF_{BCF(cor)}$ increases accuracy (F1 measure) of recommendations more than 12% compared to CF_{MJD}. The PIP similarity based CF has F1 measure close to that of the CF_{MJD}. The traditional similarity measures based CFs are worst performers and they have F1 measures less than 0.50. This shows that traditional measures could not retrieve relevant items properly.

We analyze results of the experiments conducted on the ML_2 subset. The ML_2 is ex-trimly sparse subset. Fig. 5 shows that CPC based CF outperforms existing similarity measures in both MAE and RMSE metrics. All these measures depend on the number of co-rated items. As a result, these CFs cannot improve accuracy with the increasing the number of nearest neighbors. The ML_2 subset has an average of 16 users who share only one co-rated item with an active user. However, BCF based CFs outperform all CFs in both predictive measures. The $CF_{BCF(cor)}$ attains stability with number of neighbors in the range of 140–170 ($MAE = 0.822$, $RMSE = 1.090$).

Number of successful predictions and number of perfect predictions performed by each CF are shown in Fig. 6. The proposed BCF

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

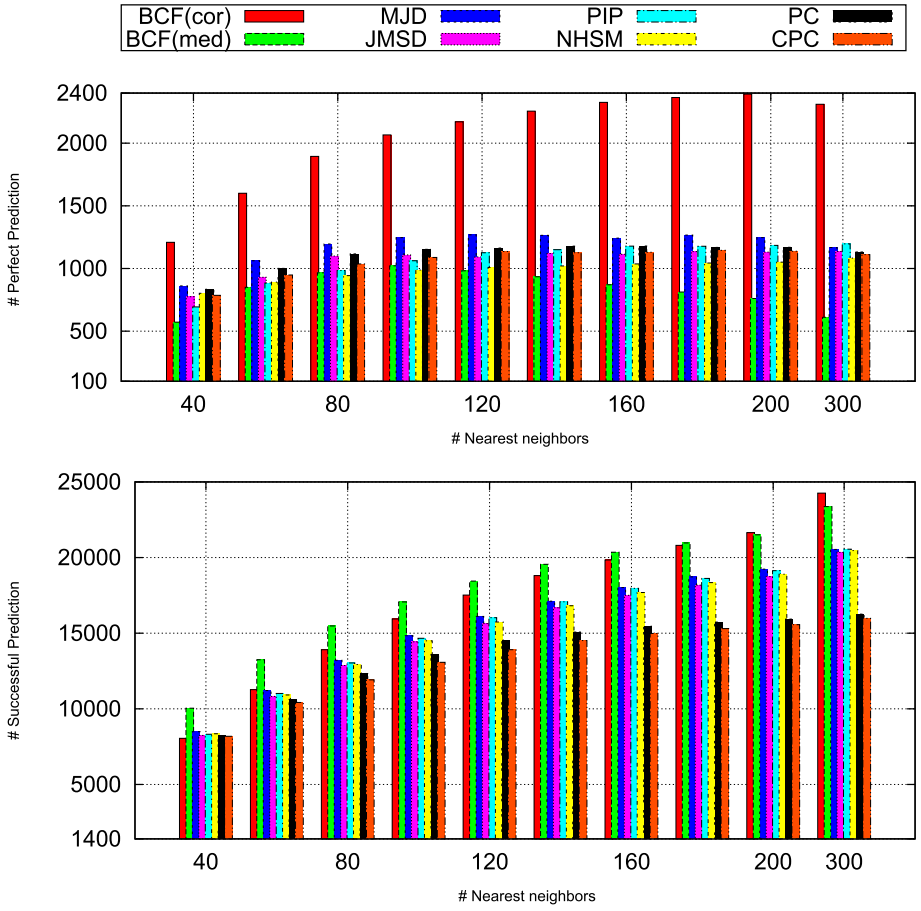

Fig. 3. Number of *perfect prediction*, *successful predictions* of CFs with increasing number of nearest neighbors on ML_1 subset (# Predictions requested = 29,665).

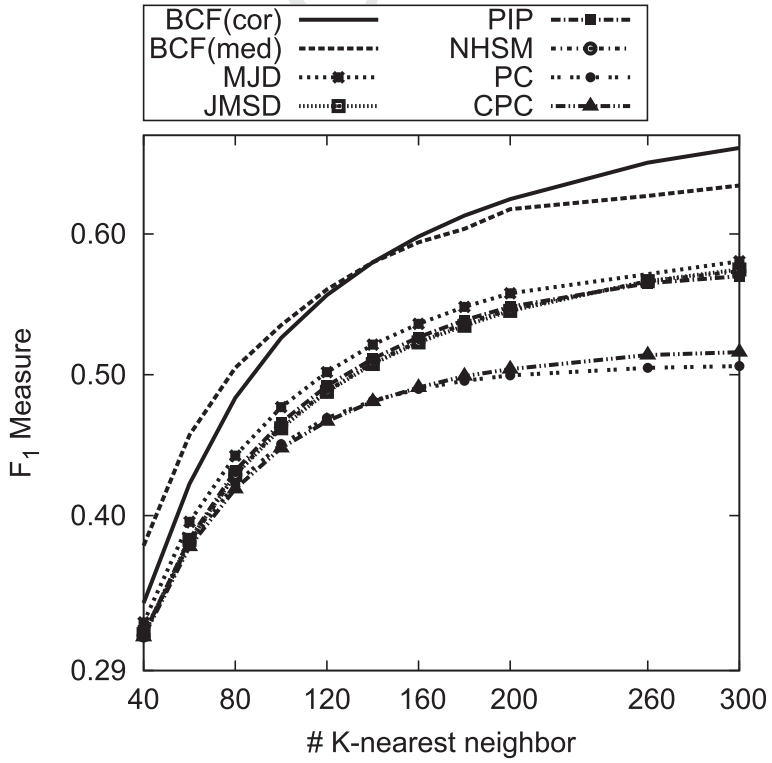

Fig. 4. F1 measure vs *K*-nearest neighbors on ML_1 subset ($\kappa = 0.18$).

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

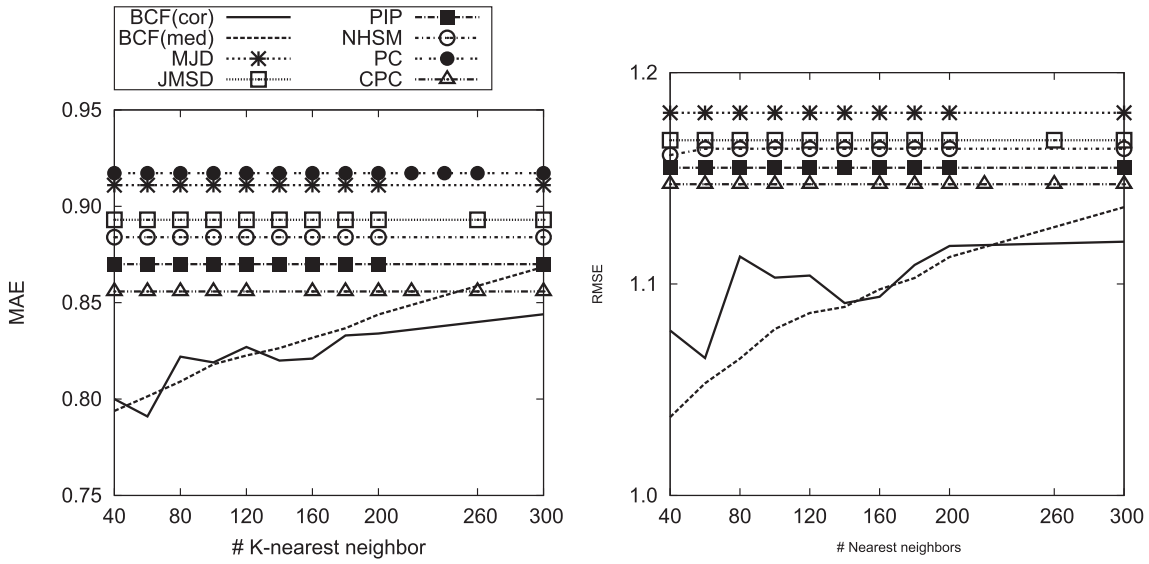

Fig. 5. *MAE*, *RMSE* measures vs *K*-nearest neighbors on ML_2 subset ($\kappa = 0.20$).

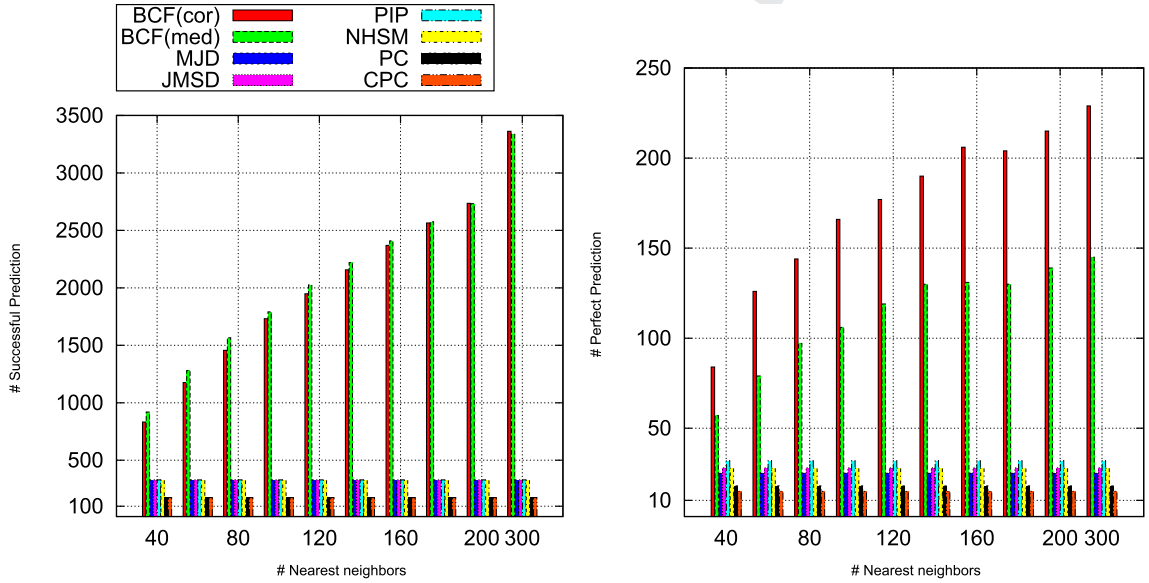

Fig. 6. Number of successful predictions vs *K*-nearest neighbors and number of perfect predictions vs *K*-nearest neighbors on ML_2 subset ($\kappa = 0.20$). Total number of predictions requested to each CF is $5 \times 1000 = 5000$.

based $CF_{BCF(cor)}$ and $CF_{BCF(med)}$ make highest and second highest number of perfect and successful predictions, respectively. Number of successful predictions, number of perfect predictions are increased with increasing the number of nearest neighbors for proposed measure. However, all other measures based CFs cannot increase neither number of successful predictions nor number of perfect predictions with increasing the number of nearest neighbors on ML_2 subset. The number of successful prediction for all these CFs is fixed at 329 out of total 5000 requested predictions (Fig. 6). The $CF_{BCF(cor)}$ has highest perfect prediction (229 out of total 3362 valid predictions) with $K = 300$. The $CF_{BCF(med)}$ is equally well for providing successful predictions. This shows that our similarity measures are effective to draw useful neighbors for an active user in highly sparse ML_2 subset and BCF does not depend on the number of co-rated items.

The F1 measures obtained after applying all CFs on ML_2 are plotted in Fig. 7, which shows that $CF_{BCF(cor)}$ and $CF_{BCF(med)}$ perform significantly better among the all CFs discussed here. The F1 measures of BCF based CFs keep on increasing with increasing the number of nearest neighbors. With an average user rating less than 7 in ML_2 subset, proposed measure can attain a F1 measure, which is close to 0.55. However, existing similarity based collaborative filter attains a F1 measure, which is less than 0.10, *i.e.*, BCF based CFs provide recommendation results four times more accurate than other approaches. The traditional similarity measures based CF_{PC} and CF_{CPC} have a F1 value of less than 0.05. This shows that traditional measures are not useful measures in providing relevant items in sparse rating data. Our measure can provide reliable recommendation after receiving few ratings from an active user. Therefore, BCF based recommender system can handle highly sparse ratings dataset.

We executed each CF on comparatively denser dataset Net_1, in which an active user on an average rates more than seven co-rated items with its neighbors (Fig. 1). The MAE and RMSE of each CF is plotted and shown in Fig. 8. The MJD based CF provides better accuracy among the existing measures including traditional

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

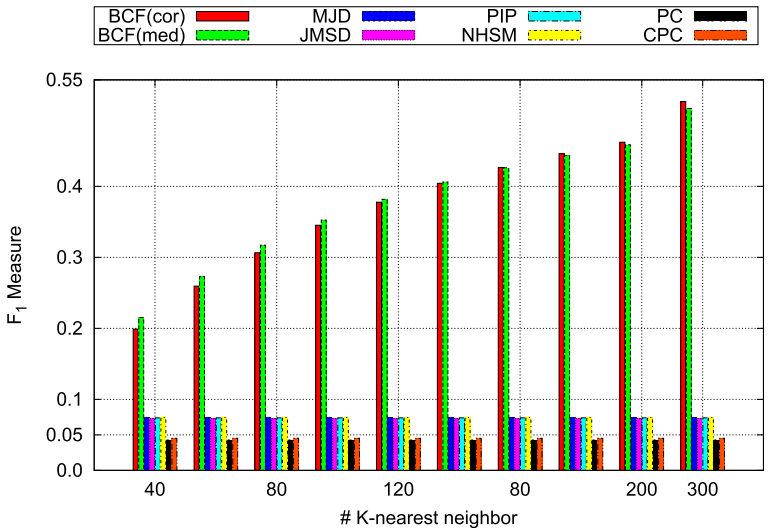

Fig. 7. F1 measure vs K-nearest neighbors on ML_2 subset.

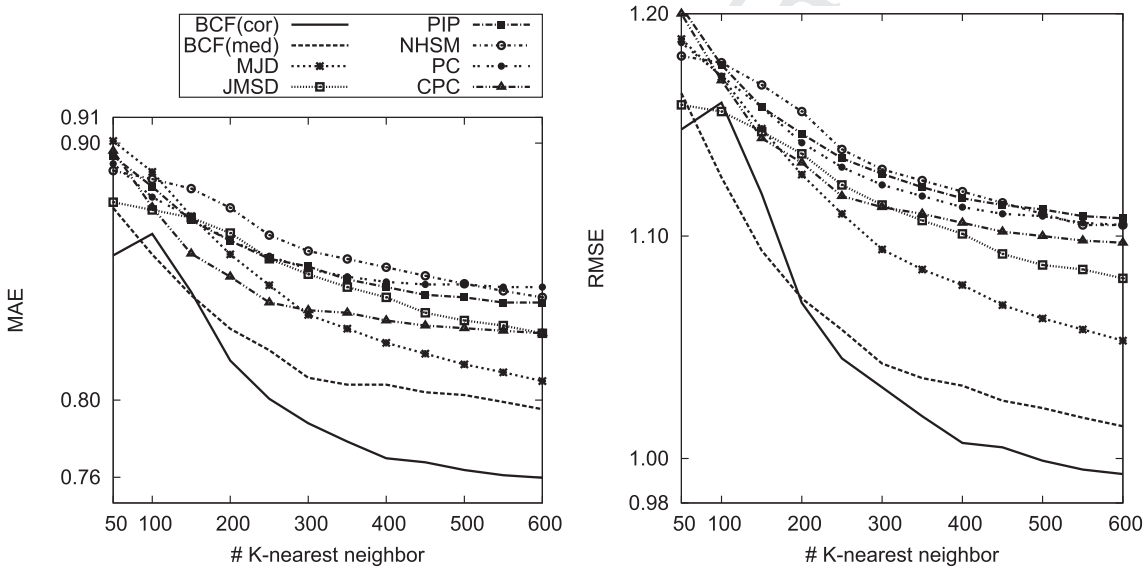

Fig. 8. *MAE*, *RMSE* measures vs K-nearest neighbors on Net_1 subset ($\kappa = 0.25$).

measures PC, CPC. The accuracy of each CF increases with increasing the number of nearest neighbor of an active user. The MJD based CF makes faster improvement compared to the other existing measures such as JMSD, PIP, NHSM, PC and CPC. However, BCF based $CF_{BCF(cor)}$ outperforms all existing measures in both accuracy metrics. The $CF_{BCF(cor)}$ attains a MAE of 0.76 and a RMSE of less than 0.99 with $K = 600$. Recommendation results produced by $CF_{BCF(cor)}$ are 5% more accurate than the results produced by MJD based CF. This is due the fact that Net_1 subset is denser (average user rating more than 21) than other subsets. Our approach works better in sparse scenario.

We tested the quality of neighborhood of an active user in terms of providing the number of successful predictions by each CF on Net_1 subset. The plot (Fig. 9) shows that BCF(med) based $CF_{BCF(med)}$ provides maximum number of successful predictions (24,089), which is close to the number of requested prediction to the system (24,408). Our proposed $CF_{BCF(cor)}$ is also successful for providing second highest number of valid predictions. The closest competitor is the PIP based CF, which makes as much as 22,322 successful predictions at $K = 600$. The traditional measures based CFs perform poorly in this metric. Number of perfect predictions by each CF is also reported in Fig. 9.

We computed F1 measure of each CF and F1 measures are plotted (Fig. 10). Our BCF measure based both CFs perform well in F1 measures in wide range of K values. The $CF_{BCF(cor)}$ is the best for recommending relevant items (F1 close to 0.68) to active users in subset Net_1.

To show effectiveness of the proposed similarity measure in sparse dataset, we computed F1 measures of all CFs after executing them on Net_2 subset ($\kappa = 0.10\%$). It is found that $CF_{BCF(cor)}$ and $CF_{BCF(med)}$ outperform all other CFs (Fig. 11). The PIP based CF and its improved version NHSM attain a F1 measure of less than 0.40, whereas $CF_{BCF(med)}$ achieves a F1 measure close to 0.58 and $CF_{BCF(cor)}$ attains a F1 measure of 0.56 with $K = 600$. Proposed BCF based CFs provide recommendation results 40% more accurate compared to the PIP, MJD based CFs. This shows the superiority

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

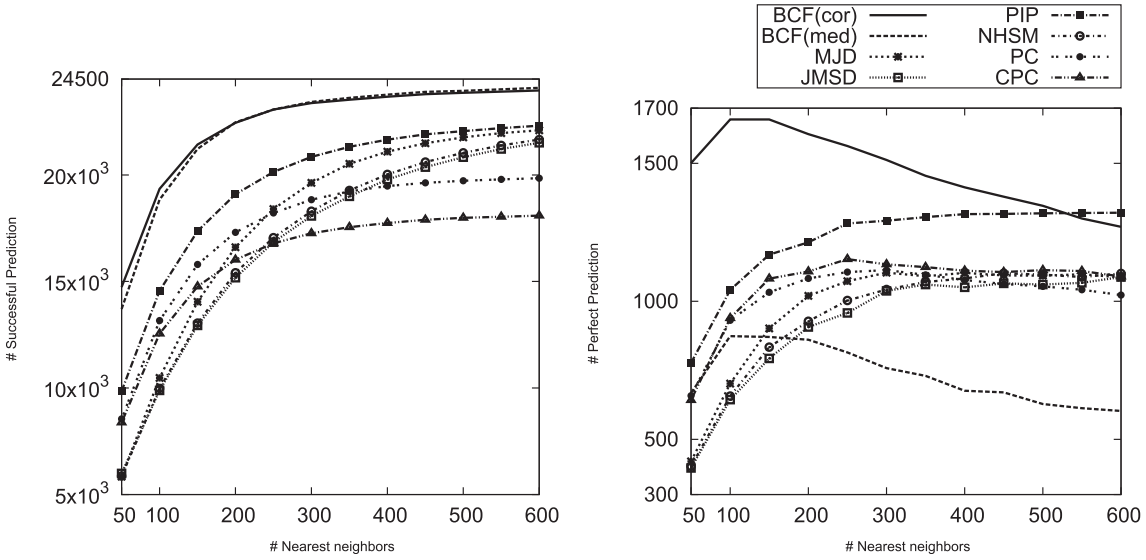

Fig. 9. Number of successful predictions vs K-nearest neighbors, number of perfect predictions vs K-nearest neighbors on Net_1 subset ($\kappa = 0.25$). Total number of predictions requested to each CF is 24,408.

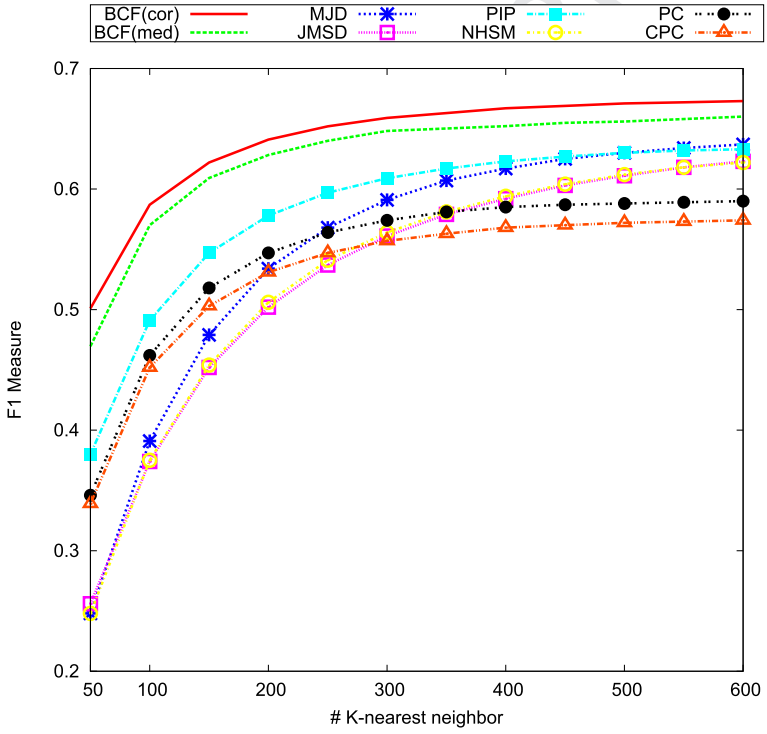

Fig. 10. F1 measure vs K-nearest neighbors on Net_1 subset ($\kappa = 0.25$).

of our proposed measures over all existing measures. The traditional measures based CFs (CF_{PC}, CF_{CPC}) attain a F1 value of less than 0.32.

The MAE and RMSE obtained from Net_2 subset are plotted and shown in Fig. 12. Traditional PC based CF produces highest MAE (Fig. 12). However, another traditional CPC based CF shows better performance than $CF_{MJD}, CF_{JMSD}, CF_{PIP}, CF_{NHSM}, CF_{PC}$ in both metrics (MAE, RMSE). It may be noted that CPC based CF makes least number of valid predictions (Fig. 13). The CPC makes 6613 times valid predictions whereas other non traditional measures (PIP, MJD) make close to 11,000 times valid predictions. The $CF_{BCF(med)}$ can perform close to 20,000 successful predictions out 24,408 requested predictions. Proposed BCF based collaborative filterings ($CF_{BCF(cor)}, CF_{BCF(med)}$) outperform all CFs including CPC based CF in both accuracy metrics. The $CF_{BCF(cor)}$ is the best and it makes lowest MAE (0.811) and RMSE (1.076) with $K = 600$. We report number of successful predictions and number of perfect predictions performed by each CF in Fig. 13 and similar trends are found.

Finally, we executed all these CFs on a subset named *YM* (Yahoo Music) with $\kappa = 0.32$. The average user rating in YM is less than 4 but average ratings per item is close to 50.

The predictive accuracy (MAE and RMSE) of these CFs are shown in Fig. 14. The plot clearly shows that BCF based CFs outperform all other measures based CFs. The $CF_{BCF(cor)}$ produces best

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

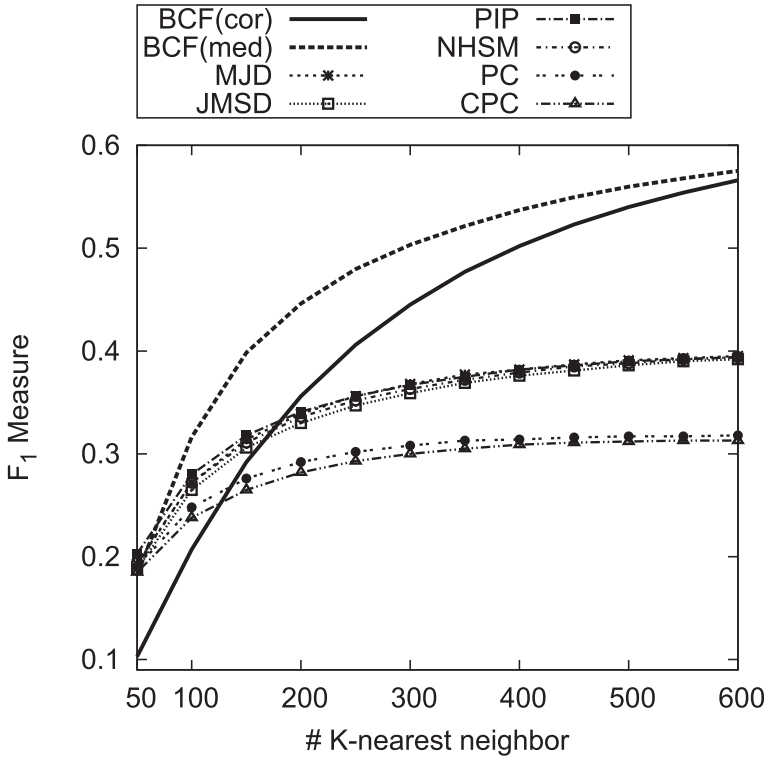

Fig. 11. F1 measure vs *K*-nearest neighbors on Net_2 subset ($\kappa = 0.10$).

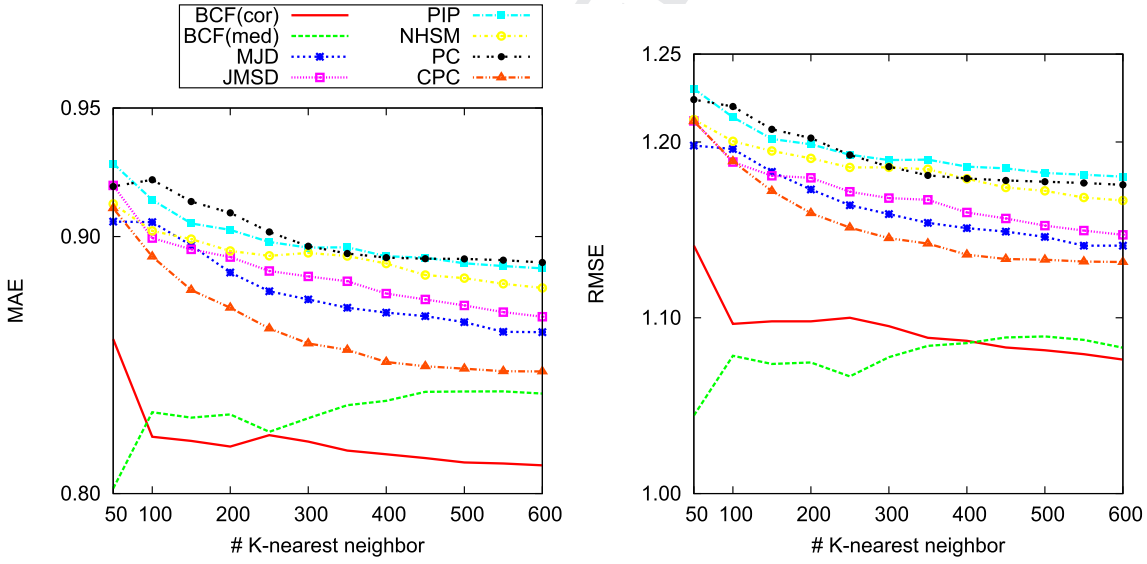

Fig. 12. *MAE*, *RMSE* measures vs *K*-nearest neighbors on Net_2 subset ($\kappa = 0.10$).

accuracy in MAE (1.21) and $CF_{BCF(med)}$ performs least RMSE (1.63) at $K = 400$. The PIP, NHSM, MJD make same error (MAE, RMSE). It can be noted that CPC based CF makes second lowest valid predictions after PC based CF. However, BCF based CFs make much higher number (18,475 at $K = 400$) of valid predictions compared to PIP, NHSM, and MJD based CFs.

The F1 measure provide qualitative importance of recommendations suggested by a CF. F1 measures of all CFs are obtained after getting executed them on YM subset. Here, we found that our proposed BCF measure based CFs provide better reliable recommendations compared to the other CFs (Fig. 15). The proposed $CF_{BCF(med)}$ provides remarkably good recommendations (F1 = 0.68) with average user user ratings less than 4 (Table 4). The $CF_{BCF(med)}$ retrieves 10% more accurate good items (relevant) compared to MJD based CF. It can be noted that CPC based CF produces second lowest F1 measure. Other BCF measure based $CF_{BCF(cor)}$ is also found to be outperforming existing measures based CFs and it attains its maximum value of 0.63 at $K = 400$.

We also computed number of successful predictions and number of perfect predictions performed by each CF. Proposed $CF_{BCF(med)}$ and $CF_{BCF(cor)}$ show similar trends in these metrics as we observed in earlier experiments on other subsets.

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

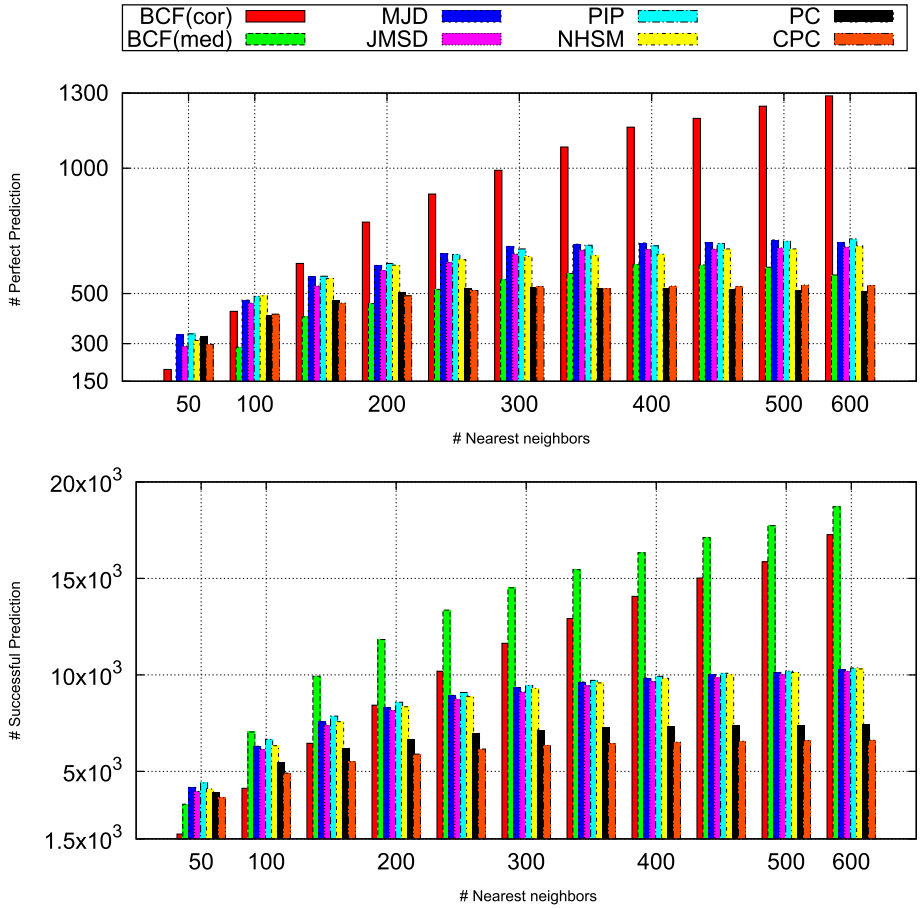

Fig. 13. Number of successful predictions vs *K*-nearest neighbors and number of perfect predictions vs *K*-nearest neighbors on Net_2 subset ($\kappa = 0.10$). Total number of predictions requested to each CF is 24,408.

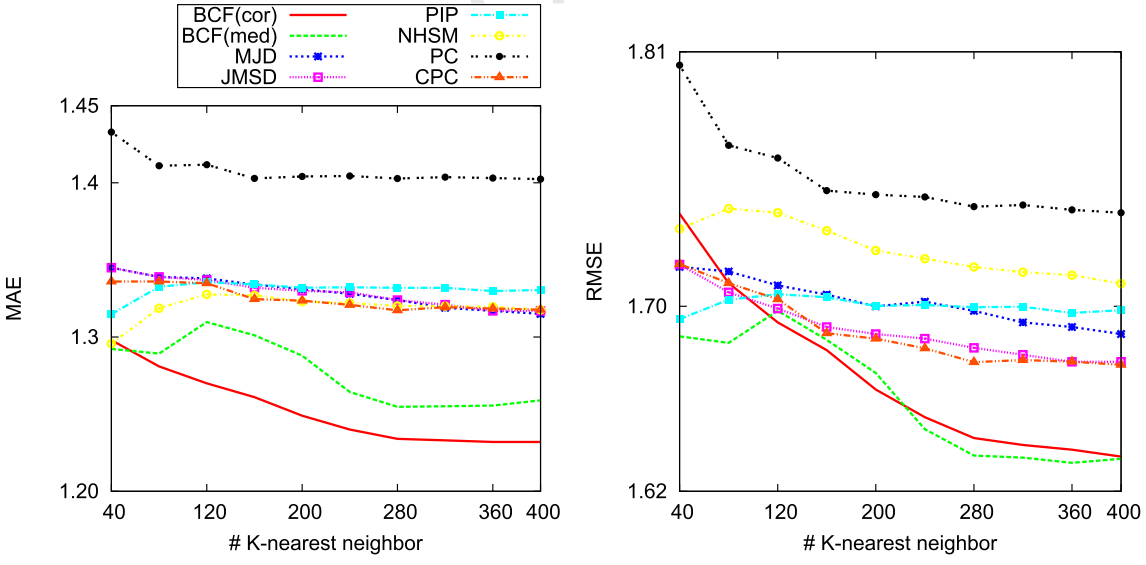

Fig. 14. *MAE*, *RMSE* measures vs *K*-nearest neighbors on YM subset ($\kappa = 0.32$).

We conducted comprehensive experiments on five subsets with various sparsity levels ($\kappa = 0.10, 0.18, 0.20, 0.25, 0.32$) to show effectiveness of proposed BCF measure for collaborative filtering in sparse data. It is found that BCF measure using both functions (Eq. (8) and Eq. (9)) outperforms existing measures in most of the accuracy metrics such as MAE, RMSE, F1 measure, number of successful predictions. However, BCF(med) based CF is found to be not effective one in providing number of perfect predictions, which is another important metric to design a successful RS. However, BCF(cor) based CF is superior over all existing measure

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001

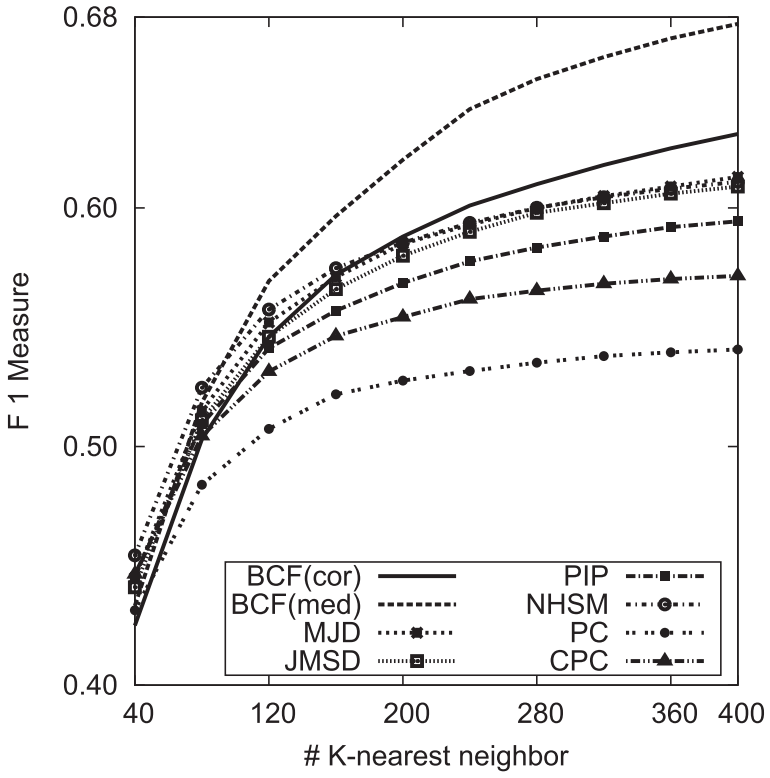

Fig. 15. $F1$ measure vs K-nearest neighbors on YM subset ($\kappa = 0.32$).

based CFs in all accuracy metrics discussed here. Therefore, we suggest to use $CF_{BCF(cor)}$ as a recommender system in a sparse dataset.

5. Conclusion and future works

The state-of-the-art similarity measures for neighborhood based collaborative filtering cannot provide reliable recommendations to the active users in sparse data as they cannot utilize full ratings information while finding neighborhood of active users. We proposed BCF measure, which utilizes all ratings information comprehensively for locating useful neighbors of an active user in sparse ratings dataset. As a result, BCF based CF can provide reliable item recommendations to active user in sparse data. The main advantage of this measure is that it does not depend on co-rated items unlike other measures. Experiments on real rating datasets show that BCF based CF can provide highly reliable recommendations in highly sparse data with few user and few item ratings.

This research can be extended in two directions. Firstly, diversity in recommendation is an important factor in developing successful recommender system. This work can be extended to provide good diversity along with reliable recommendation in sparse data. Secondly, non traditional similarity measures could be explored as the local similarity in BCF measure to increase further accuracy of the system.

Acknowledgments

This work is carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under Grant Agreement 246016.

References

[1] P. Resnick, H.R. Varian, Recommender systems, Commun. ACM 40 (3) (1997) 56–58.
[2] D. Billsus, C.A. Brunk, C. Evans, B. Gladish, M. Pazzani, Adaptive interfaces for ubiquitous web access, Commun. ACM 45 (5) (2002) 34–38.
[3] G. Linden, B. Smith, J. York, Amazon.com recommendations: item-to-item collaborative filtering, IEEE Internet Comput. 7 (1) (2003) 76–80.
[4] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, J. Riedl, MovieLens unplugged: experiences with an occasionally connected recommender system, in: Proceedings of the 8th International Conference on Intelligent user interfaces, 2003, pp. 263–266.
[5] K. Lang, NewsWeeder: learning to filter netnews, in: Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 331–339.
[6] M. Pazzani, D. Billsus, Learning and revising user profiles: the identification of interesting web sites, Mach. Learn. 27 (3) (1997) 313–331.
[7] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (6) (2005) 734–749.
[8] J. Bobadilla, F. Ortega, A. Hernando, A. Gutirrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.
[9] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artif. Intell. 2009 (2009) 4:2.
[10] D. Billsus, M.J. Pazzani, Learning collaborative information filters, in: Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp. 46–54.
[11] T. Hofmann, Latent semantic models for collaborative filtering, ACM Trans. Inf. Syst. 22 (1) (2004) 89–115.
[12] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 426–434.
[13] A. Paterek, Improving regularized singular value decomposition for collaborative filtering, in: Proceeding of KDD Cup Workshop at 13th ACM Int. Conf. on Knowledge Discovery and Data Mining, 2007, pp. 39–42.
[14] Y. Koren, Factor in the neighbors: scalable and accurate collaborative filtering, ACM Trans. Knowl. Discov. Data 4 (1) (2010) 1:1–1:24.
[15] C. Desrosiers, G. Karypis, A comprehensive survey of neighborhood-based recommendation methods, in: Recommender Systems Handbook, 2011, pp. 107–144.
[16] H. Yildirim, M.S. Krishnamoorthy, A random walk method for alleviating the sparsity problem in collaborative filtering, in: Proceedings of the 2008 ACM Conference on Recommender Systems, 2008, pp. 131–138.
[17] A. Bhattacharyya, On a measure of divergence between two statistical populations defined by their probability distributions, Bull. Calcutta Math. Soc. 35 (1) (1943) 99–109.

[18] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, 1994, pp. 175–186.
[19] K. Ali, W. van Stam, TiVo: making show recommendations using a distributed collaborative filtering architecture, in: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 394–401.
[20] M.D. Ekstrand, J.T. Riedl, J.A. Konstan, Collaborative filtering recommender systems, Found. Trends Human–Comput. Interact. 4 (2) (2011) 81–173.
[21] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, 2001, pp. 285–295.
[22] G. Karypis, Evaluation of item-based top-N recommendation algorithms, in: Proceedings of the Tenth International Conference on Information and Knowledge Management, 2001, pp. 247–254.
[23] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., New York, NY, USA, 1986.
[24] 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, 1995, pp. 210–217.
[25] J. Bobadilla, F. Serradilla, J. Bernal, A new collaborative filtering metric that improves the behavior of recommender systems, Knowl.-Based Syst. 23 (6) (2010) 520–528.
[26] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, Inf. Sci. 178 (1) (2008) 37–51.
[27] H. Luo, C. Niu, R. Shen, C. Ullrich, A collaborative filtering framework based on both local user similarity and global user similarity, Mach. Learn. 72 (3) (2008) 231–245.
[28] D. Anand, K.K. Bharadwaj, Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities, Expert Syst. Appl. 38 (5) (2011) 5101–5109.
[29] H.-N. Kim, A. El-Saddik, G. Jo, Collaborative error-reflected models for cold-start recommender systems, Decis. Support Syst. 51 (3) (2011) 519–531.
[30] J. Bobadilla, F. Ortega, A. Hernando, A collaborative filtering similarity measure based on singularities, Inf. Process. Manage. 48 (2) (2012) 204–217.
[31] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, Know.-Based Syst. 26 (2012) 225–238.
[32] K. Choi, Y. Suh, A new similarity function for selecting neighbors for each target item in collaborative filtering, Knowl.-Based Syst. 37 (2013) 146–153.
[33] H. Liu, Z. Hu, A. Mian, H. Tian, X. Zhu, A new user similarity model to improve the accuracy of collaborative filtering, Knowl.-Based Syst. 56 (2014) 156–166.
[34] B.K. Patra, R. Launonen, V. Ollikainen, S. Nandi, Exploiting Bhattacharyya similarity measure to diminish user cold-start problem in sparse data, in: Proceedings 17th International Conference Discovery Science (DS 2014), 2014, pp. 252–263.
[35] L. Zhen, G.Q. Huang, Z. Jiang, Collaborative filtering based on workflow space, Expert Syst. Appl. 36 (4) (2009) 7873–7881.
[36] L. Zhen, G.Q. Huang, Z. Jiang, Recommender system based on workflow, Decis. Support Syst. 48 (1) (2009) 237–245.
[37] L. Zhen, Z. Jiang, H. Song, Distributed recommender for peer-to-peer knowledge sharing, Inf. Sci. 180 (18) (2010) 3546–3561.
[38] F.J. Aherne, N.A. Thacker, P. Rockett, The Bhattacharyya metric as an absolute similarity measure for frequency coded data, Kybernetika 34 (4) (1998) 363–368.
[39] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR 2000), 2000, pp. 2142–2149.
[40] A.K. Jain, On an estimate of the Bhattacharyya distance, IEEE Trans. Syst. Man Cybern. SMC-6 (11) (1976) 763–766.
[41] T. Kailath, The divergence and Bhattacharyya distance measures in signal selection, IEEE Trans. Commun. Technol. 15 (1) (1967) 52–60.
[42] F. Nielsen, S. Boltz, The Burbea-Rao and Bhattacharyya centroids, IEEE Trans. Inf. Theory 57 (8) (2011) 5455–5466.
[43] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53.

Please cite this article in press as: B.K. Patra et al., A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data, Knowl. Based Syst. (2015), http://dx.doi.org/10.1016/j.knosys.2015.03.001