



# Recommendations with Sparsity Based Weighted Context Framework

Veer Sain Dixit and Parul Jain<sup>(✉)</sup>

Atma Ram Sanatan Dharam College, University of Delhi, Delhi, India  
veersaindixit@rediffmail.com,  
paruljainpj@rediffmail.com

**Abstract.** Context-Aware Recommender Systems (CARS) is a sort of information filtering tool which has become crucial for services in this big era of data. Owing to its characteristic of including contextual information, it achieves better results in terms of prediction accuracy. The collaborative filtering has been proved as an efficient technique to recommend items among all existing techniques in this area. Moreover, incorporation of other evolutionary techniques in it for contextualization and to alleviate sparsity problem can give an additive advantage. In this paper, we propose to find the vector of weights using particle swarm optimization to control the contribution of each context feature. It is aimed to make a balance between data sparsity and maximization of contextual effects. Further, the weighting vector is used in different components of user and item neighborhood-based algorithms. Moreover, we present a novel method to find aggregated similarity from local and global similarity based on sparsity measure. Local similarity gives importance to co-rated items while global similarity utilizes all the ratings assigned by a pair of users. The proposed algorithms are evaluated for Individual and Group Recommendations. The experimental results on two contextually rich datasets prove that the proposed algorithms outperform the other techniques of this domain. The sparsity measure that is best suited to find aggregation is dataset dependent. Finally, the algorithms show their efficacy for Group Recommendations too.

**Keywords:** Sparsity measure · Particle swarm optimization · Local similarity  
Global similarity · Group recommendations

## 1 Introduction

Context-Aware Recommender Systems (CARS) have gained the high attention of experts and researchers for item recommendations due to an important role played by contextual information in them [1, 2]. Dey (2001) defined the context as “Context is a piece of information that describes the circumstances of an entity”. The choice of items is usually different in different contextual situations. To quote a few examples: (a) One would choose to listen different music if weather is pleasant and/or road is free rather than it is being a hot summer day and/or traffic jam, (b) The choice of restaurant would be different if one is going on quick business lunch rather than going to dine with

girlfriend. Therefore, to improve accuracy and user satisfaction, CARS have been studied in various domains encompassing information retrieval, mobile applications, e-commerce, e-learning, management, and marketing.

The crux of CARS is to incorporate contextual information in various mathematical [1, 2, 5], probabilistic [1, 2], soft computing [1, 2, 16] and particle swarm optimization techniques [3, 7]. The recommender systems(RS) involving collaborative filtering (CF) technique are most effective and widely used among all existing RS. The quality and accuracy of such RS can be increased by using new and improved similarity measures along with utilization of contexts in an effective manner.

**Major Challenges in CARS.** There remain many challenges in CARS such as utilization of contextual information, data sparsity, scalability and cold start problem. Selection and application of contextual factors in making recommendations is clearly a blunt instrument. However, user-item-rating matrices are sparse since generally all items are not evaluated by users. Sparsity problem becomes more severe when these matrices are diluted with contextual factors. Use of too many contextual factors in algorithms increases data sparsity and few context factors fails to bring contextual effects in recommendations. In our previous research [10], we have addressed this issue by formation of context communities which are utilized in different components of the algorithm (CAWP). Although, the algorithm included the context and increased accuracy, but it is domain dependent. Optimization is the core of research. In this research, we find the optimal set of weights for the whole context vector using particle swarm optimization (PSO) to alleviate data sparsity problem.

Another major issue is to find similarity between two users and/or items, especially in context aware datasets. More the similarity measure is improved, better are the recommendations. Typically, conventional similarity measures such as Pearson Correlation Coefficient (PCC), Cosine(COS), Mean Squared Difference(MSD) can be calculated on items which are commonly rated by two users and the users who have rated common items. They ignore global rating information. Moreover, contextual information is not considered by them while finding similarity between two users and items. Even some researches have employed newly emerging similarity measures such as NHSM [18], Bhattacharya Coefficient with correlation [17] and PSS based similarity measure [18, 20] which overcome one or another problem mentioned above but do not consider contextual situation into account.

**Our Contribution.** Motivated by the above mentioned issues, the primary contribution in this work is presented as follows:

- A context weighing vector is computed using PSO to weight the contribution of all contextual features instead of context selection or relaxation. Weighting the contextual features overcome the data sparsity problem and optimizes the contextual effects. Then, these weighting vector are applied in each component of the user and item neighborhood based algorithms.
- An effective approach is proposed to combine local and global similarities based on sparsity measure. Several variations of sparsity measure are used to weight the contribution of global and local similarities. These measures caters sparse and dense data. Moreover, global similarity can be computed on non-rated items and

considers global preferences of the user behaviour while local similarity is obtained using co-rated items.

- Extending our research area, the proposed approach is evaluated for two different types of a group of users. Three different group recommendation techniques are compared to analyze the efficacy of the proposed framework. The two datasets used are contextually rich and especially designed for contextual personalization research.

The forthcoming paper is organized as follows. Few CARS related reviews and similarity measures are mentioned in Sect. 2. The detailed construction of the proposed framework and the approach used are presented in Sect. 3. Section 4 presents experimental results and their analysis. Section 5 specifies the conclusions followed by future research work.

## 2 Related Reviews

### 2.1 Context-Aware Recommendation

Context-Aware Recommender System includes context features while making a prediction. The rating estimation function is given by  $R : user \times item \times context \rightarrow rating$ . Context aware recommendation algorithm falls into three paradigms: (a) contextual pre filtering, where filtered dataset using contextual information is utilized by rating prediction algorithm, (b) contextual post filtering, where final set of recommendations are filtered using contextual information and (c) contextual modeling, where contextual information is used to predict the rating [1, 2]. The identification of valid and influential contexts is also required to be applied into recommendation algorithms [24, 26]. To identify relevant and influential context features of LDOS-CoMoDa dataset, [15] has summarized the assessments obtained from user survey and statistical testing. Another method is proposed in [22] to select optimal contexts including demographic, item and contextual features. The relevance value of each context feature set under a specific genre for IncarMusic dataset is found by [4]. A new prediction aggregation model combining predictions obtained using demographic, semantic and social contexts is demonstrated by [9]. An approach is presented by [25] to analyze several direct context prediction algorithms based on multilabel classification.

### 2.2 Sparsity Problem

Many techniques have been used by the researchers to handle data sparsity issue. It is one of the major challenges observed in this field. Especially in context aware datasets where the rating dataset is filtered with contextual information, the matrices become more sparse. DCR algorithm is proposed in [21] to handle data sparsity problem where the relaxed context constraints are used in the prediction algorithm. Another approach CAWP formed context communities and included weighted percentile method [10]. DCW for CARS [23] describes to find weights of contextual features and those users which possess context similarity greater than the threshold are used by the algorithm.

### 2.3 Similarity Measures

Traditional similarity measures such as Pearson Correlation Coefficient (PCC), Mean Square Difference (MSD), Cosine (COS), Jaccard are mostly used by recommender systems for computation of similarity between a pair of user or item. These measures have several drawbacks such as few co-rated item problems, utilization of only local rating information and non-inclusion of global ratings [1, 2, 14, 18, 20]. Hence some new similarity measures are proposed to overcome the drawbacks. A new similarity measure based on Bhattacharya Coefficient is evaluated by [14] to handle sparse data that do not depend on co-rated items. A heuristic similarity model which considers both local and global contexts of the user behaviour is experimentally shown in [18]. A model based on a mean measure of divergence is defined by [13, 20] that takes rating habits of a user into account. These measures do not consider contextual information and suffer from one or the other problem, so we attempt to form a combination of local, global and contextual similarity measure to improve accuracy.

### 3 The Proposal

This section presents construction and details of the proposed framework depicted in Fig. 1. The framework consists of Sparsity Based Weighted Context Recommendation Unit (SWCRU) and Group Recommendation Unit (GRU). The SWCRU predicts ratings via user neighborhood based and item neighborhood based algorithms. To achieve this, first, the optimum weight of different context features and genres are obtained using particle swarm optimization (PSO). The overlap metric and weights of context features are used to find contextual similarity value. This value is utilized by different parts of both user and item neighborhood based algorithms. Also, local and global similarities are found to exploit their strengths by the prediction algorithms. The sparsity measure is employed to make a balance between local and global similarity since both performs differently under different sparse data scenario. The GRU unit presents three different group recommendation techniques Merging, Multiplicative and Merging-Multiplicative for performance analysis of the proposed algorithms for a group of users. Random Groups are used for this purpose.

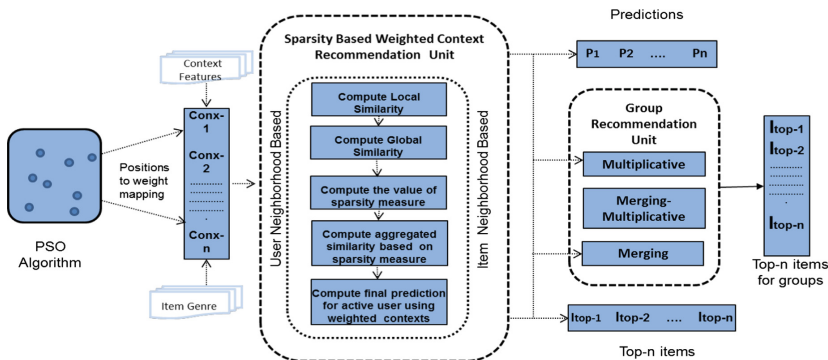


Fig. 1. The proposed framework

### 3.1 PSO to Learn Optimal Weighting Vector $w$

We assume that those rating which is more similar in contexts are more valuable in making predictions. To handle data sparsity problem, we learn the optimal weight  $w$  for each context feature using PSO instead of filtering out some context features. The  $w$  can take the weight as the real values in the range of [0,1]. These weighted values control the contribution of each context feature in the recommendation algorithms.

#### Particle Representation and Initial Population

Each context feature weight is represented with 8 binary digits in range of [0,255]. If there are ‘n’ features then 8n bits represent a particle. After reaching termination criteria, binary value of each weight is converted to its decimal equivalent [3, 7]. Then each weight is divided by the total weight to get a normalized value.

#### Particle Dynamics

PSO consists of collection of candidate solutions called swarm where each candidate solution represents a particle. These particles continuously move in search space by some velocity. The velocity and position w.r.t. each particle in every dimension is updated at each time stamp. The following rules are followed to update the swarm [7].

$$vel_i = rw * vel_i + cons_1 * r_1 (pos_{pBest,i} - pos_i) + cons_2 * r_2 (pos_{GBest} - pos_i)$$

If ( $|vel_i| > vel_{max}$ ), ( $vel_{max}/|vel_i|$ ) \*  $vel_i$ ;  $pos_i = pos_i + vel_i$

where the position of current particle  $i$  is represented by  $pos_i$ , the best position attained by particle is known by  $pos_{pBest,i}$ ,  $pos_{GBest}$  elaborates the swarm’s global best,  $vel_i$  represents the velocity of particle  $i$ ,  $rw$  is the random inertia weight lies between 0.4 and 0.9,  $cons_1$  and  $cons_2$  are spring constants whose values are set as 2.0 by empirical suggestions [22],  $r_1$  and  $r_2$  are random numbers between 0 and 1.  $vel_{min}$  and  $vel_{max}$  are  $(ul - lb)/2$ . In our case, it is tuned to 2.0 after experimental analysis. Swarm size is 10 and the number of iterations performed before termination is 20.

#### The Fitness Function

The  $i$ th particle’s fitness value in the swarm is computed using the following fitness function [3, 7].

$$Fitness_i = \frac{1}{S_R} \sum_{t=0}^{S_R} |(ar_t - pr_t)|$$

$S_R$  is the cardinality of a training set for the active user.  $ar_t$  and  $pr_t$  are actual and predicted rating of item  $t$  respectively.

#### Termination Criteria

We opted to take a specified number of iterations as terminating situation. This also means that the PSO algorithm terminates after executing specified number of iterations.

### 3.2 Overlap Metric to Find Contextual Similarity

The Overlap metric finds the similarity between two objects  $obj_i$  and  $obj_j$  and widely used for categorical attributes [19]. It is a simple and effective and is defined by

$$S(obj_i, obj_j) = \frac{\sum_{a=1}^m S_a(obj_{ia}, obj_{ja})}{m}$$

where  $a$  depicts a particular attribute and  $m$  is used to represent the total number of attributes of the object.  $obj_{ia}$  means  $a$ -th attribute of object  $obj_i$ . if  $obj_{ia} = obj_{ja}$ , then the value of  $S_a(obj_{ia}, obj_{ja}) = 1$  otherwise it is 0.

#### Illustrative Example

Using Table 1, the contextual similarity between  $user1$  and  $user4$  via Overlap metric is computed as  $S(user1, user4) = \frac{3}{5} = 0.6$ .

### 3.3 Weighted Overlap Metric

This metric is used to assess how much weight should be given to a rating  $r_{a,i,c2}$  using weighted vector  $w$  obtained using PSO. The weighted overlap metric to find context similarity between target context  $c1$  and different user context  $c2$  is given by:

$$O_w(c1, c2, w) = \frac{\sum_{func \in c1 \cap c2} w_{func}}{\sum_{func \in c1 \cup c2} w_{func}} \tag{1}$$

which means total context similarity of target user with some other user will be obtained by adding weights of those context features where the values match.

**Table 1.** Rating matrix for an item  $i$  by different users under different contextual conditions.

User	Day	Timing	Place	Companion	Mood	Rating
User1	Weekend	Night	Home	Family	Positive	4
User2	Holiday	Evening	Friend's home	Friend	Neutral	5
User3	Weekend	Afternoon	Theatre	Colleagues	Positive	4
User4	Holiday	Night	Home	Family	Neutral	3

### 3.4 Weighted Local Similarity

To calculate local similarity between two users or two items, we used Pearson Correlation Coefficient which uses commonly rated items to find the value.  $c1$  represents the context vector of  $u_x$  or  $i_x$  and  $c2$  represents the context vector of  $u_y$  or  $i_y$ .

The weighted variant to find similarity between two users,  $simu - loc_w$  is given by

$$simu - local_w(u_x, u_y, w) = \frac{\sum_{t=1}^{n'} (r_{u_x, i_t} - \bar{r}_{u_x}) - (r_{u_y, i_t} - \bar{r}_{u_y}) O_w(c1, c2, w)}{\sqrt{\sum_{t=1}^{n'} (r_{u_x, i_t} - \bar{r}_{u_x})^2 O_w(c1, c2, w)} \sqrt{\sum_{t=1}^{n'} (r_{u_y, i_t} - \bar{r}_{u_y})^2 O_w(c1, c2, w)}} \quad (2)$$

where  $i_t : t = 1, 2, \dots, n' \wedge n' \leq n$  represents a set consisting of those items which  $u_x$  and  $u_y$  had rated and  $n$  identifies the total number of accessible items.

The weighted variant to find similarity between two items,  $simi - loc_w$  is given by

$$simi - local_w(i_x, i_y, w) = \frac{\sum_{t=1}^{m'} (r_{u_t, i_x} - \bar{r}_{i_x}) - (r_{u_t, i_y} - \bar{r}_{i_y}) O_w(c1, c2, w)}{\sqrt{\sum_{t=1}^{m'} (r_{u_t, i_x} - \bar{r}_{i_x})^2 O_w(c1, c2, w)} \sqrt{\sum_{t=1}^{m'} (r_{u_t, i_y} - \bar{r}_{i_y})^2 O_w(c1, c2, w)}} \quad (3)$$

where  $\{u_x : x = 1, 2, \dots, m' \wedge m' \leq m\}$  represents a set consisting of users who have rated  $i_x$  and  $i_y$ , where  $m$  identifies the total number of accessible users.

### 3.5 Weighted Global Similarity

#### Bhattacharyya Coefficient

The Bhattacharyya Coefficient finds similarity value between two statistical samples [14]. If  $pq$  and  $pr$  be the discrete probability distributions under same domain  $D$ , the Bhattacharyya coefficient (BC) between  $pq$  and  $pr$  is given by.

$$BC(pq, pr) = \sum_{x \in D} \sqrt{pq(x)pr(x)}$$

Following it, the similarity between two users  $u1$  and  $u2$  is given as:

$$BC(u1, u2) = \sum_{k=1}^m \sqrt{(\widehat{D}_{u1k})(\widehat{D}_{u2k})} \quad (4)$$

where  $(\widehat{D}_{u1k})$  and  $(\widehat{D}_{u2k})$  represents users rating values under domain  $D$  and  $(\widehat{D}_{u1k}) = \frac{\#k}{\#u}$  where  $\#k$  = total count of items which are rated as  $k$  (value),  $\#u$  = total count of items rated by user  $u$ .

#### Illustrative Example

Consider the rating scale lies in the range  $\{1, 2, 3\}$  and user  $u1$  and  $u2$  made rating on five different items (Table 2).

The BC coefficient is calculated as:

$$BC(u1, u2) = \sum_{t=1}^3 \sqrt{(\widehat{u1}_t)(\widehat{u2}_t)} = \sqrt{\frac{2}{3} * \frac{1}{2}} + \sqrt{\frac{1}{3} * \frac{1}{2}} + \sqrt{\frac{0}{3} * \frac{0}{2}} = 0.9855$$

**Table 2.** User-item rating matrix.

User/Item	<i>i1</i>	<i>i2</i>	<i>i3</i>	<i>i4</i>	<i>i5</i>
<i>u1</i>	1	0	2	0	1
<i>u2</i>	0	1	0	2	0

**Proximity-Significance-Singularity (PSS)**

PSS similarity is used as local measure [11, 18] which punishes bad similarity and reward good similarity and is defined as follows:

$$sim(u1, u2)^{PSS} = \sum_{i \in I} PSS(r_{u1,i}, r_{u2,i}) \tag{5}$$

where *I* represents all the items rated by user *u1* and *u2* and *r<sub>u1,i</sub>* means rating assigned by user *u1* to item *i*.

$$PSS(r_{u1,i}, r_{u2,i}) = Proximity(r_{u1,i}, r_{u2,i}) \times Significance(r_{u1,i}, r_{u2,i}) \times Singularity(r_{u1,i}, r_{u2,i}).$$

Proximity considers absolute difference between two ratings and assigns penalty to disagreement.

Significance assumes that those ratings which are far off from the median are more significant.

Singularity uses difference of two ratings from the mean of their rating vector.

$$Proximity(r_{u1,i}, r_{u2,i}) = 1 - \frac{1}{1 + \exp(-|r_{u1,i} - r_{u2,i}|)}$$

$$Significance(r_{u1,i}, r_{u2,i}) = 1 - \frac{1}{1 + \exp(-|r_{u1,i} - r_{med}| * |r_{u2,i} - r_{med}|)}$$

$$Singularity(r_{u1,i}, r_{u2,i}) = 1 - \frac{1}{1 + \exp(-|(\frac{r_{u1,i} + r_{u2,i}}{2}) - \mu_i|)}$$

where  $\mu_i$  is the mean rating of item *i*.

**Hybrid Similarity Metric**

Combining the strengths of Weighted Overlap (for context similarity) stated by Eq. (1), Bhattacharya Coefficient (for global similarity) described by Eq. (4) and PSS (for local component) defined by Eq. (5), the hybrid similarity measure is given by the Eqs. (6) and (7).

$$sim_u - global_w(u_x, u_y, w) = \sum_{a \in I_{u_x}} \sum_{b \in I_{u_y}} BC(u_x, u_y) * O_w(c1, c2, w) * s(u_x, u_y)^{PSS} \tag{6}$$

$$sim_i - global_w(i_x, i_y, w) = \sum_{a \in U_{i_x}} \sum_{b \in V_{i_y}} BC(i_x, i_y) * O_w(c1, c2, w) * s(i_x, i_y)^{PSS} \tag{7}$$



### 3.6 Sparsity Measure ( $\vartheta$ )

Previous researches have verified that when the data sparsity is high then the global similarity makes more accurate predictions while in case of low sparsity, local similarity performs better. The various sparsity measures are proposed in [3] to ensure that the locally similar neighbors and globally similar neighbors should be weighted differently in different scenario of data. We propose to use these sparsity measures to get the correct proportions of local and global similarities. The various sparsity measures that utilized are as follows.

#### Overall Sparsity Measure ( $\vartheta_1$ )

It is uniform for all users and considers sparsity of entire matrix. It is computed as

$$\vartheta_1 = 1 - \left( \frac{m_R}{(m_U X m_I)} \right)$$

where  $m_R$ ,  $m_U$  and  $m_I$  represents total count of ratings in the entire matrix, total count of unique users and total count of unique items in the matrix respectively.

#### User Dependent Sparsity Measure ( $\vartheta_2$ )

The intuition behind this metric is that those users who have rated less items will not get much reliable local neighborhood. It is user specific and remains constant for all the items of the active user. The value of  $\vartheta_2$  is defined as follows.

$$\vartheta_2 = 1 - \left( \frac{m_u}{(\max_{u \in U} m_u)} \right)$$

where  $m_u$  represents number of items which user  $u$  has rated.

The forthcoming measure addresses the sparsity at user-item level since sometimes globally similar neighbors shows superiority depending on the items rated by the users.

#### Local Global Ratio ( $\vartheta_3$ )

The value of  $\vartheta_3$  is computed as:  $\vartheta_3 = 1 - \left( \frac{|L_{Neigh}(a,i)|}{|G_{Neigh}(a,j)|} \right)$

where  $L_{Neigh}(a, i)$  represents the set of locally similar neighbors of user  $a$  who have rated item  $i$  and  $G_{Neigh}(a, j)$  represents the set of globally similar neighbors of user  $a$ .  $\vartheta_4$  is defined as average of  $\vartheta_1, \vartheta_2$  and  $\vartheta_3$ .

#### Aggregated Similarity

The correct proportion of local and global similarities can achieve better quality predictions in sparse and dense data scenario. Thus, the aggregated similarity is a linear combination of local and global similarity and is defined by Equation

$$sim - agrg(i, j) = \vartheta X sim - global_w(i, j, w) + (1 - \vartheta) X sim - local_w(i, j, w) \quad (8)$$

where  $i$  represents  $u_x$  or  $i_x$  and  $j$  represents  $u_y$  or  $i_y$  depending on whether it is user or item neighborhood algorithm. The values of  $\vartheta$  can be calculated with the help of any of these sparsity measure  $\vartheta_1$ , where  $i = 1, 2, 3$  or  $4$ .

The value of  $sim - local_w(i, j, w)$  is computed by Eqs. (2) or (3) and  $sim - global_w(i, j, w)$  using Eqs. (6) or (7) depending on user or item neighborhood algorithm.

### 3.7 Predictions and Recommendations

The following neighborhood based algorithms are being used for rating prediction towards active user  $a$  towards an unrated item:

#### Weighted Context User Based Using Sparsity Measure ( $WCUB_{\vartheta-sim}$ )

$$P_{a,i,c} = \bar{w}(a, O_w(c1, c2, w)) + \frac{\sum_{t \in N_a} sim - aggr(a, t)(\omega(t, O_w(c1, c2, w)) - \bar{w}(t, O_w(c1, c2, w)))}{\sum_{t \in N_{a,k}} sim - aggr(a, t)} \quad (9)$$

#### Weighted Context Item Based Using Sparsity Measure ( $WCUB_{\vartheta-sim}$ )

$$P_{a,i,c} = \bar{w}(i, O_w(c1, c2, w)) + \frac{\sum_{t \in N_i} sim - aggr(i, t)(\omega(t, O_w(c1, c2, w)) - \bar{w}(t, O_w(c1, c2, w)))}{\sum_{t \in N_{i,k}} sim - aggr(i, t)} \quad (10)$$

**Table 3.** Description of notations used.

Notation	Description
$P_{a,i,c}$	Predicted rating user $a$ towards item $i$ in contextual situation $c$
$N_i$	Neighborhood of items that are rated by user $a$
$N_a$	Neighborhood of users who have rated item $i$
$\bar{w}(a, O_w(c1, c2, w))$	User $a$ 's average of contextually weighted ratings
$\omega(t, O_w(c1, c2, w))$	Contextually weighted rating of neighbor $t$
$sim - aggr(a, t)$	Aggregated similarity between users $a$ and $t$ after using PSO and sparsity measure
$sim - aggr(i, t)$	Aggregated similarity between items $i$ and $t$ after using PSO and sparsity measure

### 3.8 Group Recommendation Unit

This unit provides three different group recommendation techniques to evaluate the algorithms for group of users.

**Merging.** In merging, top- $n$  recommended items belonging to each member of group are merged into a single list. Then top- $n$  items of the merged list are recommended to the group [6, 8].

**Multiplicative.** In multiplicative, an aggregated value is calculated after multiplication of predicted rating obtained by each group member. Then top- $n$  items with highest value(prediction) are recommended to group as a whole [8].

**Merging-Multiplicative.** In this method, first off the top-n recommended items of each group member are merged together. Among them, top-n items are extracted. Then the new aggregated value is calculated after multiplication and the items are rearranged [8].

## 4 Experimental Evaluation

We have performed several experiments to obtain and analyze the performance of the proposed framework. The following issues are addressed:

- How do the utilization of weighted contexts via PSO in user neighborhood and item neighborhood model performs?
- To analyze the effects of sparsity measure variants controlling the contribution of local and global similarities.
- Are the proposed algorithms reliable for group of users?

### 4.1 Description, Parameter Setup and Evaluation Metrics

The experiments are conducted on two global datasets enriched with context features and especially designed for context aware personalization research. The LDOS-CoMoDa dataset is from movie domain contains 30 features and are collected from surveys [12]. IncarMusic dataset is a global dataset and collected from is [https://github.com/irecsys/CARSKit/tree/master/context-ware\\_data\\_sets](https://github.com/irecsys/CARSKit/tree/master/context-ware_data_sets) [4]. The summarized statistics of these datasets are given in Table 4.

**Table 4.** The statistics of datasets.

Datasets	# of users	# of items	# of ratings	# of contexts factors	# of user attributes	# of item attributes	Rating scale
IncarMusic	42	139	4012	8	1	8	1–5
LDOS-CoMoDa	121	1232	2296	12	4	11	1–5

For implementation purpose, those users who have given ratings to at least three items are filtered and used for experimentation. The filtered dataset is divided into three folds. Out of them one fold is utilized as test set and rest two are treated as training set. The average of five runs are presented for all measures in the results. To measure predictive accuracy, mean absolute error i.e. MAE and root mean square error i.e. RMSE are used. Also, recommended ranked list of top10 items are calculated using Precision, Recall and F1-score. For both IncarMusic and LDOS-CoMoDa data sets, an item is considered relevant (a hit) only if it is assigned a rating higher than or equal to 4 (in scale of 1–5) by the active user. Each group recommendation technique is evaluated for five Random Groups. Moreover, the experiments are performed on two sizes of groups i.e. Small Group (SG) consisting of 3–5 users and Large Group (LG) consisting of 6–8 users. Group recommendations are measured for five runs and average using F1-score (metric) is presented as result.

## 4.2 Compared Methods

The experimental results shown below are compared with three more approaches to analyze the performance of the proposed algorithms presented in Sect. 3. We choose one context aware recommendation approach *CAWP* from our previous researches [10], second DCR via BPSO [21, 23] and DCW via PSO [23] from the same domain of research.

**CAWP.** In our previous research work, we tried to come out from the dilemma of context selection by forming context communities and used a weighted percentile method to increase the accuracy [10]. We implemented the concept in user neighborhood model  $CAWP_{UB-ER}$  and item neighborhood model  $CAWP_{IB-ER}$ . We are using the best cases i.e. 90th percentile in case of movie dataset and 70th percentile in music dataset for comparison.

**DCR via BPSO.** Binary particle swarm optimization (BPSO) uses vectors of binary values to represent the position of particle instead the real-valued vectors. BPSO has been successfully demonstrated as efficient non linear optimizer for feature selection. It is available in open source libraries to understand and implement [23]. Moreover, DCR via BPSO is described [23] as the best technique to filter the context features.

**DCW via PSO.** Instead of selecting few features, DCW includes the contribution of each contextual feature which is weighted. PSO optimizes the position of particles which represents a weighting vector for context features. It is also found that PSO based algorithms outperforms genetic algorithms which is also used for the same purpose. Hence it is used in CF technique [23].

## 4.3 Results and Analysis

This section presents and discusses the experimental results of the proposed framework using LDOS-CoMoDa and IncarMusic datasets.

### Method Comparisons

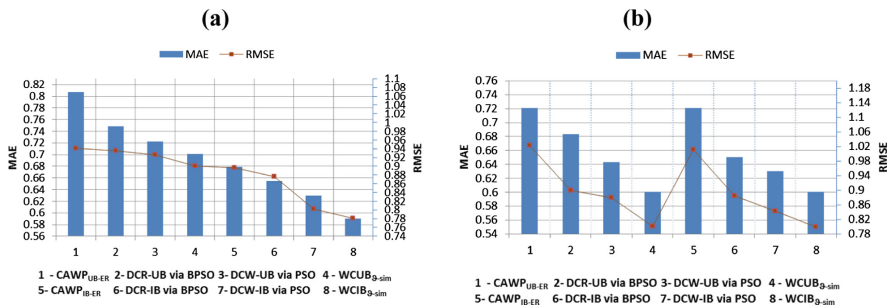
Table 5 presents the results of the proposed sparsity based weighted context recommendation technique and other context aware implementations. It is also shown in Table 5 that the proposed algorithms whether it is user or item neighborhood based outperforms the other techniques of this area. The reason could be that usage of optimum weight for context features and sparsity dependent contribution of local and global neighbors are taken. It is worth to be noted that the proposed algorithms consider global similarities too (i.e. neighbors who have not rated common items) which other compared techniques don't do.

Further, Fig. 3 illustrates a comparison in the predictive accuracy of the recommendation system using only local similarity and the combination(local and global similarities with best case of 9). Using both datasets, the combination of local and global similarities perform better than local similarities. It also verifies the assumption that two users can be similar even if they do not rate common items.

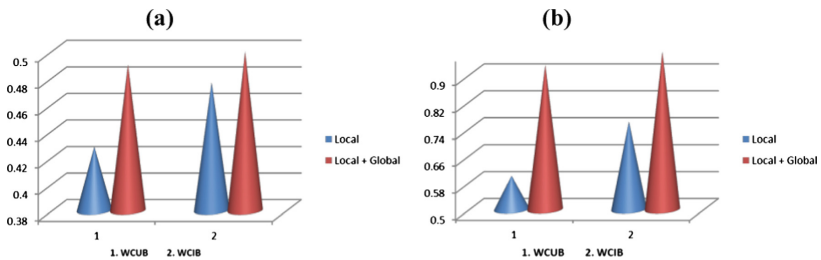
Figure 2 (a) and (b) depict that the two variants of proposed method (i.e. user neighbourhood based and item neighbourhood based) show a significant difference in

**Table 5.** The computed values of MAE, RMSE, Precision, Recall and F1-score for different algorithms using two datasets.

Datasets	IncarMusic dataset					LDOS-CoMoDa dataset				
Algorithm	MAE	RMSE	Precision	Recall	F1-score	MAE	RMSE	Precision	Recall	F1-score
CAWP <sub>UB-ER</sub> (Baseline)	0.7211	1.0231	0.6110	0.3210	0.4208	0.8073	0.9412	0.8388	0.4495	0.5853
DCR via BPSO (Baseline)	0.6832	0.8992	0.6893	0.3005	0.4185	0.7489	0.9362	0.8578	0.5901	0.6992
DCW via PSO (Baseline)	0.6431	0.8791	0.7388	0.3465	0.4717	0.7224	0.9255	0.8976	0.6589	0.7599
WCUB <sub>9-sim</sub> (Proposed)	0.6008	0.8012	0.7599	0.3471	0.4765	0.7008	0.9009	0.8999	0.6595	0.7612
CAWP <sub>IB-ER</sub> (Baseline)	0.7210	1.0114	0.6806	0.3830	0.4902	0.6794	0.8968	0.9296	0.9287	0.9291
DCR via BPSO (Baseline)	0.6502	0.8839	0.6818	0.3718	0.4812	0.6543	0.8765	0.9619	0.9218	0.9414
DCW via PSO (Baseline)	0.6301	0.8432	0.6842	0.3781	0.4870	0.6292	0.8023	0.9732	0.9556	0.9643
WCIB <sub>9-sim</sub> (Proposed)	0.6001	0.8002	0.6979	0.3892	0.4997	0.5901	0.7802	0.9798	0.9579	0.9687



**Fig. 2.** Comparison of different algorithms w.r.t. MAE and RMSE values using: (a) IncarMusic dataset (b) LDOS-CoMoDa dataset.



**Fig. 3.** Prediction accuracy (F1-score) using local similarity vs. local + global similarity on: (a) IncarMusic dataset (b) LDOS-CoMoDa dataset.

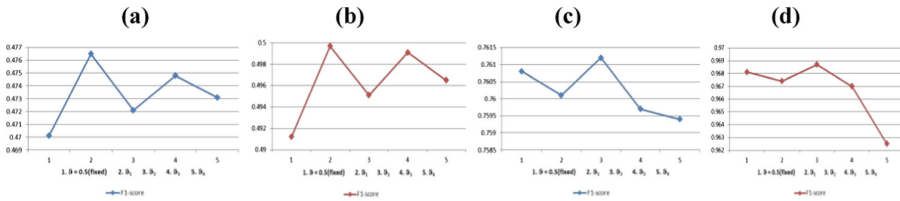
terms of MAE and RMSE. Similar trend is seen in F1-score (Table 5). Proposed algorithms reduces MAE and RMSE values remarkably compared to other baselines. The search space for BPSO is limited since the value in particle position can switch between 0 and 1 i.e. either the context feature is included or it is not. In PSO, search space becomes unlimited since the value can be in range [0,1] i.e. all context feature take some value in the range [0,1].

Hence, it can be concluded that proposed algorithms which finds optimum weight for context features using PSO and utilize the combination of local and global similarities are the best performing one. Also, item neighbourhood based algorithm are better than user neighbourhood based algorithm.

**Sensitivity Analysis of Sparsity Measure  $\vartheta$**

Figure 4 shows the comparison of results w.r.t. F1-score using different sparsity measures  $\vartheta_i$  where  $i = 1, 2, 3, 4$  or  $5$ . The prediction accuracy of  $\vartheta_2$  based on F1-score is better than others using movie dataset. The reason might be that small local neighborhood is formed since it is user specific sparsity measure. It can be observed that the similar trend is shown by both user and item neighborhood algorithms. It is worth to be noted that in IncarMusic dataset  $\vartheta_1$  performs better as dataset is comparatively less sparse and rich set of local neighbors are obtained.

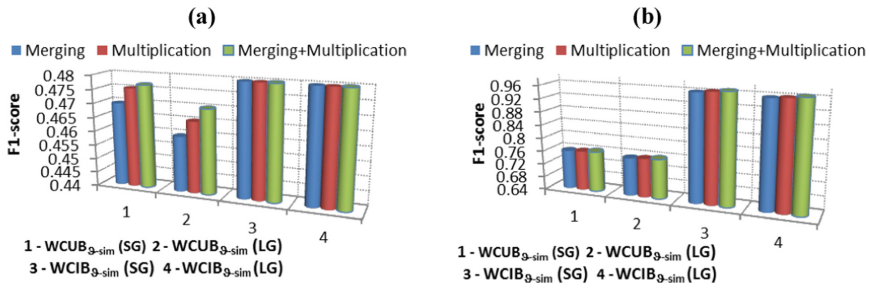
Hence, we claim that sparsity measure  $\vartheta$  can handle local and global similarities in more effective way and the choice of sparsity measure best suited, is dataset dependent.



**Fig. 4.** Sensitivity of sparsity measure  $\vartheta$  on F1-score for the datasets (a) IncarMusic in user neighborhood (b) IncarMusic in item neighborhood (c) LDOS-CoMoDa in user neighborhood (d) LDOS-CoMoDa in item neighborhood.

**Performance for Group Recommendations**

Figure 5(a) and (b) illustrates that the Merging + Multiplicative technique is slightly better than other grouping techniques. The reason could be that it is able to fix up errors more. Particularly, with user neighborhood algorithms on music dataset. F1-score values in Fig. 5 and Table 3 also reveals the effectiveness of proposed algorithms for group recommendations.



**Fig. 5.** Performance of different Group Recommendations techniques in terms of F1-score. SG represents small group and LG represents large group. User neighborhood and item neighborhood algorithms are compared on two datasets (a) IncarMusic (b) LDOS-CoMoDa.

## 5 Conclusions and Future Work

Through this paper, we tried to alleviate the data sparsity problem and the issue of finding similarity in the absence of co-rated items for context aware recommender systems. To attain these goals, we have proposed a novel framework which utilizes contextually weighted collaborative filtering techniques based on user and item neighborhood model. The aggregated similarity measure used by these algorithms attempts to take sparsity based contribution of local and global similarities to produce better quality predictions. Global similarity measure provides better predictions when data sparsity is high and can handle user rating behaviour. Local similarity measure performs well in case of low data sparsity. The algorithms are evaluated under different levels of sparsity. Also, PSO technique is used to find weights of different context features to be utilized by different components of the algorithms. Assigning weights to contextual features rather than selection or matching also solves data sparsity problem. The experimental results show that the balanced contribution of local and global similarities produce better accuracy than considering only local similarity. Moreover, PSO is an efficient optimizer for weighting context. Hence, the proposed algorithms increase predictive accuracy. Furthermore, the variant of sparsity measure that suits best is dataset dependent. The proposed algorithms are reliable for group recommendations also.

In future, we aim to utilize fuzzy logic to better understand the results.

## References

1. Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A.: Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst. (TOIS)* **23**(1), 103–145 (2005). <https://doi.org/10.1145/1055709.1055714>
2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, Paul B. (eds.) *Recommender Systems Handbook*, pp. 217–253. Springer, Boston, MA (2011). [https://doi.org/10.1007/978-0-387-85820-3\\_7](https://doi.org/10.1007/978-0-387-85820-3_7)

3. Bakshi, S., Jagadev, A.K., Dehuri, S., Wang, G.: Enhancing scalability and accuracy of recommendation systems using unsupervised learning and particle swarm optimization. *Appl. Soft Comput.* **15**, 21–29 (2014). <https://doi.org/10.1016/j.asoc.2013.10.018>
4. Baltrunas, L., et al.: InCarMusic: context-aware music recommendations in a car. In: Huemer, C., Setzer, T. (eds.) *EC-Web 2011*. LNBIIP, vol. 85, pp. 89–100. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-23014-1\\_8](https://doi.org/10.1007/978-3-642-23014-1_8)
5. Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: Context relevance assessment and exploitation in mobile recommender systems. *Pers. Ubiquit. Comput.* **16**(5), 507–526 (2012). <https://doi.org/10.1007/s00779-011-0417-x>
6. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendation with rank aggregation and collaborative filtering. In: *RecSys 2010 Proceedings of the Fourth ACM Conference on Recommender Systems*, pp. 119–126. ACM, New York (2010). <https://doi.org/10.10145/1864708.1864733>
7. Choudhary, P., Kant, V., Dwivedi, P.: Handling natural noise in multi criteria recommender system utilizing effective similarity measure and particle swarm optimization. In: *Seventh International Conferences on Advances in Computing Communications-2017*, pp. 853–862 (2017). *Procedia Computer Sciences* 115. <https://doi.org/10.1016/j.procs.2017.09.168>
8. Christensen, I.A., Schiaffino, S.: Entertainment recommender systems for group of users. *Expert Syst. Appl.* **38**, 14127–14135 (2011). <https://doi.org/10.1016/j.eswa.2011.04.221>
9. Dixit, V.S., Jain, P.: A proposed framework for recommendations aggregation in context aware recommender systems. In: *8th International conference on Cloud Computing, Data Science & Engineering*. IEEE, Noida (2018, paper accepted and presented, in press)
10. Dixit, V.S., Jain, P.: Weighted percentile based context aware recommender systems. In: *1st International Conference on Signals, Machines and Automation, AISC*. Springer, Heidelberg (2018, paper accepted and presented, in press)
11. Katpara, H., Vaghela, V.B.: Similarity measure for collaborative filtering to alleviate the new user cold start problem. In: *Third International Conference on Multidisciplinary Research and Practice*, vol. 4, no. 1, pp. 233–238 (2016)
12. Kosir, A., Odic, A., Kunaver, M., Tkalcic, M., Tasic, Jurij, F.: Database for contextual personalization. *Elektrotehniški vestnik*, vol. 78, no. 5, str. 270–274, ilustr. (2011). [English print ed.]
13. Liu, H., Hu, Z., Mian, A., Tian, H., Zhu, X.: A new user model to improve the accuracy of collaborative filtering. *Knowl.-Based Syst.* **56**, 156–166 (2014). <https://doi.org/10.1016/j.knosys.2013.11.006>
14. Miao, Z., Zhao, Z., Huang, L., Yu, P., Qiao, Y., Song, Y.: Methods for improving the similarity measure of sparse scoring based on the Bhattacharyya measure. In: *International Conference on Artificial Intelligence: Techniques and Applications* (2016). <https://doi.org/10.1016/j.eswa.2011.04.221>
15. Odic, A., Tkalcic, M., Tasic, J.F., Kosir, A.: Relevant context in a movie recommender system: users opinion vs. statistical detection. In: *Proceedings of the 4th International Workshop on Context-Aware Recommender Systems*. Dublin, Ireland (2012)
16. Panniello, U., Tuzhilin, A., Gorgoglione, M.: Comparing context-aware recommender systems in terms of accuracy and diversity. *User Model. User-Adap. Inter.* **249**(1–2), 35–65 (2014). <https://doi.org/10.1007/s11257-012-9135-y>
17. Patra, B.K., Launonen, R., Ollikainen, V., Nandi, S.: A new similarity measure using Bhattacharyya coefficient for collaborative filtering in sparse data. *Knowl.-Based Syst.* **82**, 163–177 (2015). <https://doi.org/10.1016/j.knosys.2015.03.001>
18. Saranya, K.G., Sudha Sadasivam, G.: Modified heuristic similarity measure for personalization using collaborative filtering technique. *Appl. Mathe. Inf. Sci.* **1**, 307–315 (2017). <https://doi.org/10.18576/amis/110137>



19. Sulc, Z., Režankova, H.: Evaluation of recent similarity measures for categorical data. In: 17th Application of Mathematics and Statistics in Economics, International Scientific Conference, Poland (2014). <https://doi.org/10.15611/amse.2014.17.27>
20. Wang, Y., Deng, J., Gao, J., Zhang, P.: A hybrid user similarity model for collaborative filtering. *Inf. Sci.* **418–419**, 102–118 (2017). <https://doi.org/10.1016/j.ins.2017.08.008>
21. Zheng, Y., Burke, R., Mobasher, B.: Differential context relaxation for context-aware travel recommendation. In: Huemer, C., Lops, P. (eds.) *EC-Web 2012*. LNBP, vol. 123, pp. 88–99. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-32273-0\\_8](https://doi.org/10.1007/978-3-642-32273-0_8)
22. Zheng, Y., Burke, R., Mobasher, B.: Optimal feature selection for context-aware recommendation using differential relaxation. In: *Conference Proceedings of the 4th International Workshop on Context-Aware Recommender Systems*, Dublin, Ireland. ACM RecSys (2012). <https://doi.org/10.13140/2.1.3708.7525>
23. Zheng, Y., Burke, R., Mobasher, B.: Recommendation with differential context weighting. In: Carberry, S., Weibelzahl, S., Micarelli, A., Semeraro, G. (eds.) *UMAP 2013*. LNCS, vol. 7899, pp. 152–164. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-38844-6\\_13](https://doi.org/10.1007/978-3-642-38844-6_13)
24. Zheng, Y., Burke, R., Mobasher, B.: The role of emotions in context aware recommendation. In: *Decisions@RecSys Workshop in Conjunction with the 7th ACM Conference on Recommender Systems*, Hong Kong, China, pp. 21–28. ACM (2013)
25. Zheng, Y., Burke, R., Mobasher, B.: Context recommendation using multilabel classification. In: *IEEE/WIC/ACM International Joint Conference on Web Intelligence (WI) and Intelligent Agent Technologies (IAI)*, ACM Recsys, pp. 301–304. ACM, Silicon Valley (2014)
26. Zheng, Y.: A revisit to the identification of contexts in recommender systems. In: *20th International Conference on Intelligent Users Interfaces*, ACM IUI, Atlanta, GA, USA, pp. 109–115 (2015)