Received: April 18, 2022. Revised: June 8, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Recommender System using Distributed Improved Predictive Framework with Matrix Factorization and Random Forest

Kavitha Muruganantham[1]* **Subbaiah Shanmugasundaram[2]**

[1]*PG & Research Department of Computer Science and Applications,*
Vivekanandha College of Arts and Sciences for Women, Tiruchengode, Tamilnadu, India
[2]*Department of Computer Science, Sri Krishna Arts and Science College, Coimbatore, Tamilnadu, India*
* Corresponding author's Email: kavithamphd@gmail.com

Abstract: Online digital marketing achieves their revenue according to their advertisements or sales assignment when companies have the profitable attention for recommending their products to customers via ranking them. Online customers are not able to guarantee that the items delivered through the recommendation by big data are either comprehensive or applicable to their essentials. In the past few years, recommendation frameworks were broadly applied to analyze the massive amount of data. Among those, a Distributed Predictive model with Matrix factorization and random Forest (DPMF) has achieved high efficiency to predict the item ratings. However, it accounts only for user preferences and opinions whereas the other contextual data are necessary for enhancing the efficiency of rating prediction. In this article, a Distributed Improved Predictive model with a Matrix factorization and random Forest (DIPMF) framework is proposed that considers the elements of social context and the dynamic characteristic of every user for every item to enhance the quality of prediction. The primary aim is to combine the information from the preferences, opinions and social context of each user. The social context of users is multiple features of the context such as differences in the current opinion with earlier opinion, behavior, relationship and interaction. Since each user is connected through relations and interactions. At first, the training dataset is split into an optimal amount of splits for accelerating the parallel and distributed training process. Then, the training process is carried out by the DPMF with the Distributed Improved Predictive model with Matrix factorization-Improved variant (DIPMI) to create the representation of every user's preferences, opinions and social contexts in the training set. Further, the prediction of rating is formulated as a regression challenge and solved via the Random Forest (RF) algorithm that predicts the customer's rating behavior with their opinions and social context for every product. Finally, the experiments are conducted on trip advisor and Amazon datasets to evaluate the efficiency of DIPMI and DIPMF compared to the state-of-the-art recommendation frameworks. The findings exhibit that the DIPMF on the trip advisor dataset achieves an average of 0.6826 Root Mean Square Error (RMSE), 0.5925 Mean Absolute Error (MAE), 0.8369 recommendation quality and 0.0023 Confidence Range (CR) 95% compared to the other frameworks. Similarly, the DIPMF on Amazon dataset achieves an average of 0.7591 RMSE, 0.5704 MAE, 0.8298 recommendation quality and 0.0032 CR 95% compared to the other frameworks.

Keywords: Big-data, Distributed computing, DPMF, Rating behavior, Recommendation system, Social context.

1. Introduction

In Big Data systems, the problem is to discover and scrutinize huge data for extracting the appropriate information required for certain objectives [1, 2]. Also, it increases the necessity for effective data processing that supports customers to discover suitable products such as files, films and songs [3, 4]. So, recommendation frameworks are developed which offer users adapted ideas according to their previous preferences and interest [5]. Examples of real-life applications are social networks, e-government, e-commerce, etc. One of the most effective methods used for constructing recommendation systems is Collaborative Filtering (CF). The CF approaches are split into two types:

model and memory-based [6, 8]. The model-based approaches afford suggestions according to the numerical strategies. These approaches estimate the unrecognized reviews via collecting highly identical customers or products, accordingly [9, 12].

Now, the fast growth in the number of customers, products and other data has made serious problems for standard recommendation frameworks. The essential for examining the customer's choices and interests is to create a basic framework for processing the huge volume of data. Many recommendation systems have verified better efficiency for fewer amounts of data, but they are complex to be implemented in the big-data framework [13, 14]. It processes prices at a considerable interval for varying data sizes. Data sparsity is a crucial challenge since it affects the efficiency of recommendations. Therefore, designing high-level recommendation frameworks involves attention to various problems like dealing with the data sparsity, lessening the computation period, enhancing the recommendation efficiency and managing the massive amount of data effectively. To combat all these concerns, a Distributed Predictive Model (DPM) for personalized recommendations, DPM according to the representation of the products i.e., an Improved variant of DPM (DPMI) and DPMF have been developed according to the data splitting mechanism and a new training task [15]. Also, a technique was suggested according to the distributed Matrix Factorization (MF) and RF for improving the overall efficiency. As well, this system was parallelized by the Apache Spark platform. On the contrary, this DPMF system considers only user preferences and opinions whereas additional contextual information is needed to enhance the quality of prediction.

Hence in this article, a DIPMF framework is proposed that considers the elements of social context and the dynamic behavior of every user for each item to improve the prediction quality. The main intent of this DIPMF framework is to fuse the details from the user preferences, user opinions and social context. The social context of users is many attributes of context like variation in current opinion with past opinion, behavior, relationship and interaction. Normally, users are linked via relations and interactions. So, by considering the social context of users into their preferences and opinions, the prediction quality can be enhanced. Initially, the training dataset is split into an optimal amount of splits for accelerating the parallel and distributed training process. After, the training task is performed by the distributed predictive MF frameworks with DIPMI to create the representation for every user preference, opinion and social context in the training set. Moreover, the prediction of rating is formulated as a regression challenge and solved via the RF algorithm that predicts the customer's rating attitude using their reviews and social context for every product.

The rest of the article is prepared as follows: Section 2 surveys the works related to this research work. Section 3 explains the proposed methodology and Section 4 portrays its efficiency. Section 5 summarizes the research work.

2. Literature survey

A Cache Block-matrix MapReduce (CBMR) with product-based CFs was suggested [16] to predict the user ratings. This algorithm has different main phases: determining the rating matrix and item-item concurrence matrix, creating the cache files, determining the item-item similarity matrix and prediction matrix. An Approximate Parallel Recommendation Algorithm (APRA) was proposed [17] according to Spark for handling a huge amount of data. Here, the modified behavior principle of every customer and the random sampling was adopted. Keyword-Aware Recommendation using Implicit Feedback (KAR-IF) [18] was applied on products to solve the cold-start customers. The rating estimation and suggestion were established by the two server-side units with significant processes.

A Covering Algorithm using Quotient space Granularity evaluation on Spark (CA-QGS) was proposed [19] to recommend an online facility precisely. A Pairwise Association Rule-based Recommender Algorithm (PARRA) was developed [20] that constructs a model of collective preferences autonomously of personal user interests and does not need a complicated system of ratings. An Ensemble Co-training Recommender (ECoRec) system was proposed [21] that use two or more recommender algorithms were used. A new Imputation-based Singular Value Decomposition for Recommendation (ISVDR) [22] was proposed which uses an adjacent choice algorithm according to the relationship between customers and products.

A Distributed Group Recommendation based on Extreme Gradient Boosting (DGR-EGB) algorithm [23] was suggested for dealing effectively with the curse of dimensionality challenge, identifying the groups of users and enhancing the prediction quality. An enhanced ride-sharing framework [24] using Support Vector Machine (SVM) was presented to predict the behavior of new riders by learning the client's opinion after completing their trip. A novel intelligent recommender system was designed [25]

that merges CF with the K-means clustering for recommending items to an active user. Deep auto-encoders for a multi-criteria recommender system [26] were developed. A new system was developed [27] to predict rating based on the combination of Dual Deep Learning and the Probabilistic MF (DDL-PMF). The DDL includes Stacked Denoising Auto Encoder (SDAE) and Long Short Term Memory (LSTM).

2.1 Problem definition

From the literature survey, the problems in the recommendation systems are:

- The efficiency was less due to the limited ratings of different users for variety of items.
- The recommendation relies on the social relationship interactions, which should be considered to improve the accuracy.
- In some researches, user preferences and opinions only considered, while more features like context, etc., are required to increase the recommendation accuracy.

2.2 Research objective

This research focuses on increasing the accuracy of recommendation system by extracting the context features of users for variety of items. To achieve this, the DIPMF framework is proposed, which involves the different primary phases. The functions in those phases are briefly described in below section.

3. Proposed methodology

This section explains the DIPMF framework in detail. The schematic representation of DIPMF is portrayed in Fig. 1. This DIPMF has three major phases:

- *Data partition phase:* First, the training dataset is split into an optimal amount of splits to speed up the parallel and distributed system.
- *Training phase:* Second, the training process is executed using the distributed predictive MF frameworks with DIPMI to estimate the representation of preferences, opinions and social context of each user in the training set.
- *Prediction phase:* Third, the prediction of rating is formulated as a regression challenge and solved via the RF to predict the customer's rating attitude using their opinions and social context for every product.

Table 1 presents the notations used in this study.

Table 1. Notations used in this Study

Symbols	Description
RDD_{train}	Training data
N_p	Group of promising amount of splits
$Time(RDD_{train}, n_p)$	Minimum computation time
n_p^*	Optimal amount of splits
k	Number of the user rating behavior
$RDD_{train}^{(1)}$	Split
$X \in \mathbb{R}^{\lvert X \rvert}$	Input
$A \in \mathbb{R}^{\lvert X \rvert}$	Appropriate representation
u	User
c	Social context
i	Product
n	Amount of ratings
$R_{u,c}$	Group ratings defined by u and their c
R_i	Group ratings give for i
$V_{u,c} \in \mathbb{R}$	Rating attitude of u with c
$V_i \in \mathbb{R}$	Choices of u about i
λ_i	Variable
λ_i^*	Optimized variable
$D(x) \in \mathbb{R}$	Predicted range to enhance the prediction of u's reviews
$q^{(i)}(x)$	Objective
O	Group of promising values for approximating $D(x)$
$min(Q^{(1)})$	Minimum value of $Q^{(1)}$
$max(Q^{(1)})$	Maximum value of $Q^{(1)}$
$min(Q^{(2)})$	Minimum value of $Q^{(2)}$
$max(Q^{(2)})$	Maximum value of $Q^{(2)}$
$r_{u,c,i}$	Ground-truth rating allocated via u and c to i
$\hat{r}_{u,c,i}$	Expected rating of for c and i
$\beta_{u,c}, \gamma_{u,c}, \omega_{u,c}^*$	Variables of the learned framework
RDD_{DIPMI}	RDDs of the estimated parameters
$\{(x_j, y_j), j = 1, \dots, \lvert N \rvert\}$	RDD_{train} which is collected of $\lvert N \rvert$ instances
N	Amount of non-zero customer-product rating R
$x_j \in \mathbb{R}^{\beta+1}$	Attributes of a data j
$y_j \in \mathbb{R}$	Class of a data j
β	Amount of MFs
$L(\cdot)$	Ratings by the learned frameworks
$(E_{u,c}^{(1)} W_i^{(1)})$, $(E_{u,c}^{(2)} W_i^{(2)})$	Latent factors computed by the first and second MF frameworks
Q	Quality
RP_i	Score of i with a cost of P
Z	Total cost of i selected by u

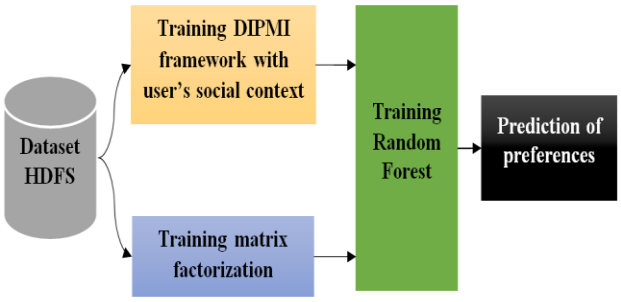

Figure. 1 Schematic representation of DIPMF framework

3.1 DIPMI: distributed improved predictive model based on fine-tuning the representation of items

DIPMI is personalized for optimizing the parameters of a distributed predictive framework by fine-tuning the estimated representations of users' opinions for every item according to the user's preferences and their social context. The social contexts are different in the current opinion from earlier opinions, behavior, relationship and interaction. Users are linked through relations and interactions. The relationships are stable links between more users. There are different kinds of relations like friendship or belonging to a similar group. Few kinds of relations are undirected or mutual whereas others are directed or asymmetrical. Interactions occur while a user communicates with others. The interaction categories are direct messages, replies and user remarks. Interactions include the mean amount of ratings per item, a ratio of ratings with remarks and the mean amount of re-ratings per item. Most of these categories encompass the formation of ratings/opinions.

When a user shares a new rating or opinion, a source relation between the user and the review is created. The new opinion is associated with the earlier opinion or other users. So, users interact with the current opinion via responding to it, liking it, storing it, etc. Also, the behavior indicates the properties of the user's rating behavior like variation of rating times during a period and uniformity of rating.

3.1.1. Data partition

The data sharing among users is significant to the performance of big-data systems. The main intent is to split the training data RDD_{train} into an optimum amount of splits that facilitate the DIPMIF framework for accelerating the parallel and distributed training process. Consider N_p is the group of the promising amount of splits and $Time(RDD_{train}, n_p)$ is a function that denotes the computation period needed for training based on the variable n_p. The challenge is described as:

$$n_p^* = \underset{n_p}{\operatorname{argmin}} \left(Time(RDD_{train}, n_p) \right), \forall n_p \in N_p$$

$$\text{s.t. } RDD_{train} = \left(RDD_{train}^{(1)} \cup \ldots \cup RDD_{train}^{(n_p^*)} \right) \quad (1)$$

in Eq. (1), n_p^* denotes the optimal amount of splits, $RDD_{train}^{(1)}$ is the split and $Time(RDD_{train}, n_p)$ is the minimum computation period.

3.1.2. Training phase

DIPMF framework is denoted as a directed acyclic graph. Every node is a function that considers the input data and generates the outcome which is then applied as input to the next node and so on. Each input $X \in \mathbb{R}^{|X|}$ is mapped to the appropriate representation $A \in \mathbb{R}^{|X|}$ as:

$$A_{u,c} = T(X = R_{u,c}) = \left(r_{u,c,i_1}, \ldots, r_{u,c,i_{|I|}} \right)$$

$$A_i = T(X = R_i) = \left(r_{u_1,c_1,i}, \ldots, r_{u_{|U|},c_{|C|},i} \right)$$

$$\text{s.t. } \forall r_{u,c,i} \in R_{u,c}, \quad r_{u,c,i} \in A_{u,c}$$

$$\forall r_{u,c,i} \in R_i, \quad r_{u,c,i} \in A_i \quad (2)$$

In Eq. (2), X is either the group of ratings $R_{u,c}$ defined by the user u and their social context c or the group of ratings R_i given for a product i. The aim of $T(X)$ is to pick every rating $r_{u,c,i} \in X$. The resultant representations $A_{u,c}$ and A_i are aggregated based on every u with c and i as:

$$V_{u,c} = S(A_{u,c}) = \frac{\sum_{r_{u,c,i} \in A_{u,c}} r_{u,c,i}}{|R_{u,c}|}$$

$$V_i = S(A_i) = \frac{\sum_{r_{u,c,i} \in A_i} r_{u,c,i}}{|R_i|} \quad (3)$$

In Eq. (3), $V_{u,c} \in \mathbb{R}$ simplifies the rating attitude of u with c and $V_i \in \mathbb{R}$ estimates the choices of u about i. To generate more precise optimized predictions, DIPMF considers that the significant description may return the reviews of u for every i. So, this is essential to consider the optimized variables of DPM and evaluate the reviews of u for updating the predicted description regarding every i and c. The aim is to adjust the below objective:

$$h = \sum_{u\in U}\sum_{c\in C}\sum_{i\in I}\left(r_{u,c,i}\left(\frac{2\left((1-\beta_{u,c})(V_{u,c}+\omega^*_{u,c})+\beta_{u,c}(V_i+\lambda_i)\right)}{\gamma_{u,c}}\right)\right)^2 \quad (4)$$

The partial derivative of h regarding the variable λ_i is described as:

$$\frac{\partial h}{\partial \lambda_i} = 2\sum_{r_{u,c,i}\in R_i}\left(r_{u,c,i}\left(\frac{2\left((1-\beta_{u,c})(V_{u,c}+\omega^*_{u,c})+\beta_{u,c}(V_i+\lambda_i)\right)}{\gamma_{u,c}}\right)\right)\left(\frac{-2\beta_{u,c}}{\gamma_{u,c}}\right) \quad (5)$$

From Eq. (5), $\lambda_i \in \mathbb{R}$ is calculated as:

$$\lambda_i = H = \frac{\sum_{r_{u,c,i}\in R_i}\frac{2\beta_{u,c}}{\gamma_{u,c}}\left(r_{u,c,i}\left(\frac{2\left((1-\beta_{u,c})(V_{u,c}+\omega^*_{u,c})+\beta_{u,c}V_i\right)}{\gamma_{u,c}}\right)\right)}{\sum_{r_{u,c,i}\in R_{u,c}}\left(\frac{2\beta_{u,c}}{\gamma_{u,c}}\right)^2} \quad (6)$$

For every i, the optimized variable $\lambda_i^* \in \mathbb{R}$ is described as:

$$\lambda_i^* = \lambda_i + D(x) \quad (7)$$

In Eq. (7), $D(x) \in \mathbb{R}$ is the predicted range used for enhancing the prediction of u's reviews. The aim of λ_i^* is to compute the best description of reviews of u for i and c.

To compute the most suitable $D(x)$, DIPMI applies the below objective:

$$q(x) = [q^1(x), q^2(x)] \quad (8)$$

$$q^{(1)}(x) = \frac{\sum_{r_{u,c,i}\in R_i}\left|r_{u,c,i}-\frac{2\left((1-\beta_{u,c})(V_{u,c}+\omega^*_{u,c})+\beta_{u,c}(V_i+\lambda_i^*)\right)}{\gamma_{u,c}}\right|}{|R_i|} \quad (9)$$

$$q^{(2)}(x) = \sqrt{\frac{\sum_{r_{u,c,i}\in R_i}\left(r_{u,c,i}\left(\frac{2\left((1-\beta_{u,c})(V_{u,c}+\omega^*_{u,c})+\beta_{u,c}\left(V_i+\lambda_i^*\right)\right)}{\gamma_{u,c}}\right)\right)^2}{|R_i|}} \quad (10)$$

Here, $\left(q^{(i)}(x), i = 1,2\right)$ denote two objectives. So, the result that fulfills the optimal trade-off is devised as:

$$D(x) = \underset{x}{\operatorname{argmin}}\left(\frac{q^{(1)}(x)-min(Q^{(1)})}{max(Q^{(1)})}\right) + \left(\frac{q^{(2)}(x)-min(Q^{(2)})}{max(Q^{(2)})}\right), \forall x \in O \quad (11)$$

$$Q^{(1)} = \left(q^{(1)}(x): x \in O\right) \quad (12)$$

$$Q^{(2)} = \left(q^{(2)}(x): x \in O\right) \quad (13)$$

Here, O is the group of promising values for approximating $D(x)$, $min\left(Q^{(1)}\right)$ and $max\left(Q^{(1)}\right)$ are the least and highest value of $Q^{(1)}$, accordingly.

3.1.3. Prediction phase

For the approximated variables of the DIPMI, the expected rating of u for c and i is described as:

$$\hat{r}_{u,c,i} = \frac{2\left((1-\beta_{u,c})(V_{u,c}+\omega^*_{u,c})+\beta_{u,c}(V_i+\lambda_i^*)\right)}{\gamma_{u,c}} \quad (14)$$

In Eq. (14), $\beta_{u,c}, \gamma_{u,c}, \omega^*_{u,c}$ and λ_i^* indicate the variables of the learned framework. A basic motivation of Eq. (14) is to predict the rating $\hat{r}_{u,c,i}$ via taking into account the optimized description of rating attitude for u including their reviews for i and c.

Algorithm

Input: RDD_{train} : Training data in HDFS, n_p^* : Optimal amount of splits, k: Number of the user rating behavior

Output: RDD_{DIPMI} : RDDs of the estimated parameters

Begin

Split RDD_{train} into n_p^* splits $RDD_{train} = \left(RDD_{train}^{(1)} \cup \ldots \cup RDD_{train}^{(n_p^*)}\right)$;

Merge $R_{u,c}$ provided by every $u \in U$;

Merge R_i provided for every $i \in I$;

for $(every\ u \in U)$

Estimate $V_{u,c}$of u;

end for

for $(every\ i \in I)$

Estimate V_i about i;

end for

$RDD_{DIPMI} < -DIPMI\left(RDD_{train}, n_p^*, k\right)$

for $(every\ subproblem_i)$

Compute λ_i according to $\frac{\partial h}{\partial \lambda_i} = 0$ using Eq. (6);

Compute λ_i^* using Eq. (7);

end for

$RDD_{DIPMI} < -DIPMI\left(\left(i_1, \lambda^*_{i_1}\right), \ldots, \left(i_N, \lambda^*_{i_N}\right)\right)$;
Return RDD_{DIPMI};
End

3.2 DIPMF: distributed improved predictive framework with matrix factorization and random forest

A key goal of DIPMF is designed by accounting for the benefit of DIPMI with MF and RF frameworks to enhance the efficiency of suggestion. Also, every known rating $r_{u,c,i} > 0$ in RDD_{train} is characterized by attributes and class. After that, the prediction of rating is solved as the regression challenges. Consider $N = \{(x_j, y_j), j = 1, \ldots, |N|\}$ is the RDD_{train} which is collected of $|N|$ instances and $|N|$ stands for the amount of non-zero customer-product rating R. Every $x_j \in \mathbb{R}^{\beta+1}$ and $y_j \in \mathbb{R}$ represents the attributes of a data j (i.e., created description) and the class (i.e., ground-truth rating), accordingly.

Consider β is the amount of MFs, the primary aim is to train β MF with DIPMI and then the learned frameworks are applied for creating the representation for every preference $r_{u,c,i}$ in RDD_{train} as:

$$L(r_{u,c,i}) = (x_j, y_j)$$

$$x_j = \left(E^{(1)}_{u,c} W^{(1)}_i, \ldots, E^{(\beta)}_{u,c} W^{(\beta)}_i, \hat{r}_{u,c,i}\right)$$

$$y_j = r_{u,c,i} \tag{15}$$

Here, $L(\cdot)$ is dedicated to characterizing the ratings by the learned frameworks, $\left(E^{(1)}_{u,c} W^{(1)}_i\right)$ and $\left(E^{(2)}_{u,c} W^{(2)}_i\right)$ are the latent factors computed by the first and second MF frameworks, accordingly. The basic hypothesis of Eq. (15) is to create the description which defines every choice and manipulate these descriptions by considering the RF that may provide better outcomes. After, the prediction of the rating process is resolved via RF using the pre-defined classes. The RF is a collection of decision trees that are trained according to the bagging. Once RF is trained, the learned framework is used for predicting unknown preferences.

4. Experimental results

In this section, the DIPMF framework is executed in MATLAB 2017b and its performance is evaluated with the KAR-IF [18], PARRA [20], ECoRec [21], ISVDR [22], DGR-EGB [23], SVM [24], CF-K-means [25], DDL-PMF [27], DPM [15], DPMI [15], DPMF [15] and DIPMI frameworks. Here, the items from the Trip Advisor and Amazon datasets are considered to reorganize and suggest the items to the users according to their rating behavior. To analyze the efficiency of these frameworks, MAE, RMSE, Quality (Q) and CR are utilized.

4.1 Dataset description

- *Trip Advisor Dataset:* It is acquired from the University of California-Irvine (UCI) and has 2 datasets: car and hotels ratings. The car dataset contains the complete rating of car models for 2007, 2008 and 2009. For each model year, almost 250 various cars and nearly 42230 ratings are involved. The structure of this dataset includes car brand, year, amount of ratings, power, interior, exterior, design, efficiency, quality, serviceability, pleasure and total reviews. The hotel dataset contains the complete ratings of restaurants in 10 various locations and nearly 700 restaurants are found in each location. So, it has about 259000 ratings. The structure of this dataset includes the restaurant's ID, name, website, address, locality, country, zip code, amount of ratings, neatness, accommodation, facility, price, affordability and total reviews.
- *Amazon Dataset:* It comprises 143.7 million ratings of products covering between May 1996 and July 2014. The subcategories include articles, TVs, electronics, movies, fashion, appliances, etc. In this experiment, only the movies & TV subcategory is chosen because of the high processing time for analyzing the whole dataset. Every Amazon subclass dataset has 2 different subcategories:
  1. The review set includes the reviewer's ID, name, product ID, review text, product rating, summary and time of the rating.
  2. The metadata includes product ID, name, cost, website of a product photo, related products, sales order details, model and the product types.

These datasets are handled via extracting the significant data and the ratings from specific customers to products in various years.

4.2 Evaluation metrics

- *RMSE and MAE:* Typically, RMSE and MAE are used for computing the prediction accuracy. These are represented as:

$$RMSE = \sqrt{\frac{\sum_{u\in U, c\in C, i\in I}\left(r_{u,c,i}-\hat{r}_{u,c,i}\right)^2}{n}} \quad (16)$$

$$MAE = \frac{\sum_{u\in U, c\in C, i\in I}\left|r_{u,c,i}-\hat{r}_{u,c,i}\right|}{n} \quad (17)$$

In Eqs. (16) and (17), n denotes the number of ratings, $r_{u,c,i}$ is the ground-truth rating allocated via u and c to i and $\hat{r}_{u,c,i}$ is the expected rating. The minimum error value specifies better prediction accuracy.

- *Quality (Q):* It measures the recommendation quality by

$$Q = \sum_{i=1}^{Z} RP_i \quad (18)$$

In Eq. (18), RP_i denotes the score of i with a cost of P if i is in the upper level or 1 if i is in the lower level and Z denotes the total cost of i selected by u. The highest Q value indicates the greatest recommendation quality.

- *Confidence interval Range (CR):* It is the interval around $\hat{r}_{u,c,i}$ where $r_{u,c,i}$ lies a fixed confidence level i.e., 95%. The minimum prediction interval indicates the maximum confidence of the rating prediction. It is used for measuring the rating confidence.

4.3 RMSE

The RMSE for the different frameworks using the Trip Advisor dataset with the number of splits are portrayed in Fig. 2. It indicates the RMSE of DIPMF is less than the other frameworks while the number of splits in the training set is high. For the trip advisor dataset with 40 splits, the RMSE of DIPMF is 26% less than KAR-IF, 25.4% less than PARRA, 24.8% less than ECoRec, 24.1% less than ISVDR, 23.7% less than DGR-EGB, 23% less than the SVM, 21.7% less than the CF-K-means, 21.1% less than the DDL-PMF, 20.4% less than the DPM, 18.4% less than the DPMI, 14.6% less than the DPMF and 6.5% less than the DIPMI frameworks.

Fig. 3 depicts the RMSE values for the Amazon dataset. It observes the RMSE of DIPMF is reduced than the other frameworks for rating prediction when the amount of splits in the training dataset is increased. For the Amazon dataset with 40 splits, the RMSE of DIPMF is 16.5% less than KAR-IF, 15.5% less than PARRA, 14.9% less than ECoRec, 14.4% less than ISVDR, 13.7% less than DGR-EGB, 13.2% less than the SVM, 11.8% less than the CF-K-means, 11.3% less than the DDL-PMF, 10.7% less than the DPM, 8.6% less than the DPMI, 6.6% less than the DPMF and 4% less than the DIPMI frameworks.

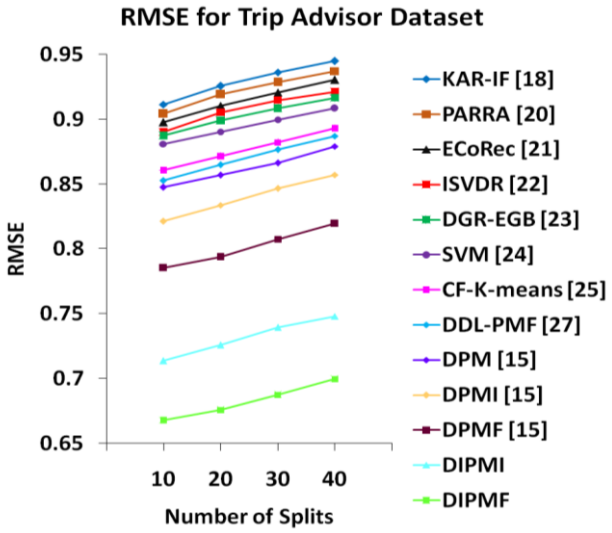

Figure. 2 RMSE vs. no. of splits for trip advisor dataset

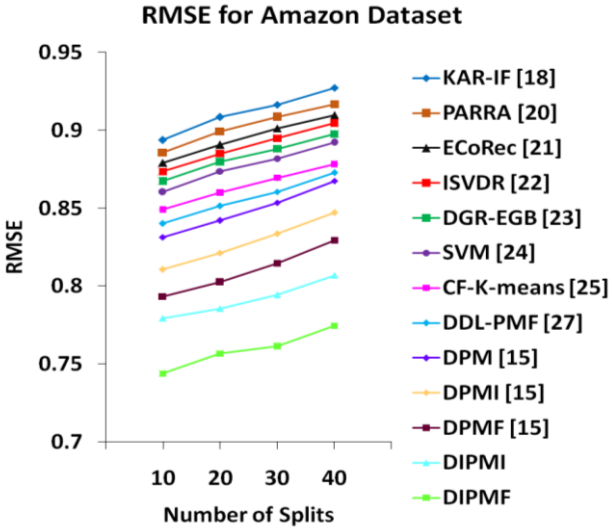

Figure. 3 RMSE vs. no. of splits for amazon dataset

4.4 MAE

The MAE values for the different frameworks using the Trip Advisor dataset with the number of splits are shown in Fig. 4. It analyzes the MAE of DIPMF is minimized than the other rating behavior prediction frameworks while increasing the number of splits in the training dataset. For the trip advisor dataset with 40 splits, the MAE of DIPMF is 18.9% less than KAR-IF, 18.2% less than PARRA, 17.7% less than ECoRec, 16.8% less than ISVDR, 16.4% less than DGR-EGB, 15.5% less than the SVM, 13.9% less than the CF-K-means, 13.2% less than the DDL-PMF, 12.8% less than the DPM, 11.7%

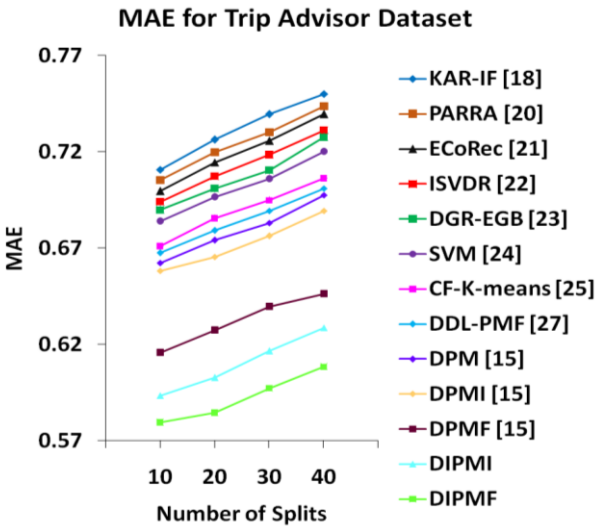

Figure. 4 MAE vs. no. of splits for trip advisor dataset

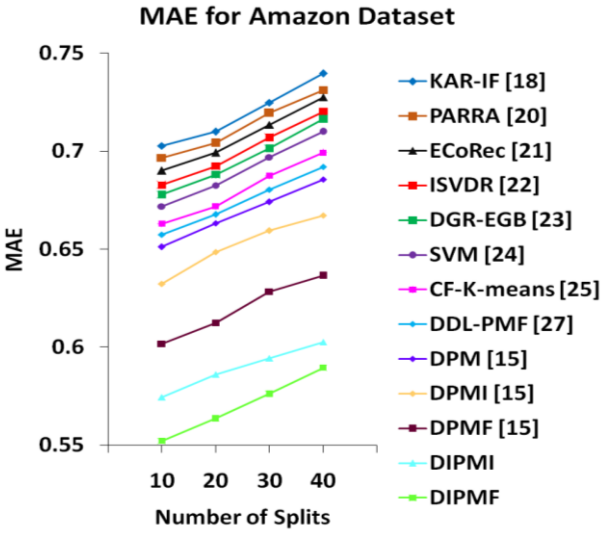

Figure. 5 MAE vs. no. of splits for amazon dataset

less than the DPMI, 5.8% less than the DPMF and 3.2% less than the DIPMI frameworks.

Likewise, Fig. 5 portrays the MAE for the Amazon dataset. It shows the MAE of DIPMF is decreased than the other frameworks for predicting the rating behavior of users while increasing the number of splits in the training dataset. For the Amazon dataset with 40 splits, the MAE of DIPMF is 20.3% less than KAR-IF, 19.4% less than PARRA, 19% less than ECoRec, 18.1% less than ISVDR, 17.7% less than DGR-EGB, 17% less than the SVM, 15.7% less than the CF-K-means, 14.8% less than the DDL-PMF, 14% less than the DPM, 11.7% less than the DPMI, 7.4% less than the DPMF and 2.2% less than the DIPMI frameworks.

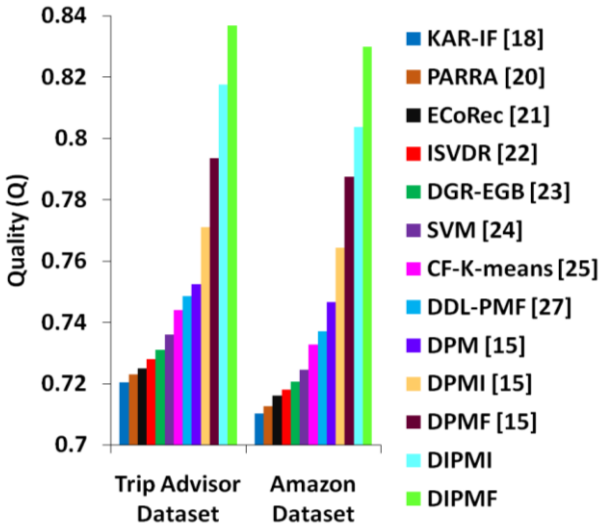

Figure. 6 Quality vs. different datasets

4.5 Quality

The Q values for the different recommendation frameworks using both the Trip Advisor and Amazon datasets are depicted in Fig. 6. It indicates the Q of DIPMF using both datasets is increased than the other frameworks. For the trip advisor dataset, the quality of DIPMF is 16.2% higher than the KAR-IF, 15.8% higher than the PARRA, 15.4% higher than the ECoRec, 15% higher than the ISVDR, 14.5% higher than the DGR-EGB, 13.7% higher than the SVM, 12.5% higher than the CF-K-means, 11.8% higher than the DDL-PMF, 11.2% higher than the DPM, 8.5% higher than the DPMI, 5.5% higher than the DPMF and 2.4% higher than the DIPMI frameworks. Similarly, for the Amazon dataset, the quality of DIPMF is 16.8% higher than the KAR-IF, 16.4% higher than the PARRA, 15.9% higher than the ECoRec, 15.6% higher than the ISVDR, 15.2% higher than the DGR-EGB, 14.5% higher than the SVM, 13.2% higher than the CF-K-means, 12.6% higher than the DDL-PMF, 11.2% higher than the DPM, 8.6% higher than the DPMI, 5.4% higher than the DPMF and 3.2% higher than the DIPMI frameworks.

4.6 Confidence range

Fig. 7 exemplifies the 95% CR values for different recommendation frameworks using both the Trip Advisor and Amazon datasets. It notices the 95% CR of DIPMF using both datasets is reduced than the other frameworks.

For the trip advisor dataset, the 95% CR of DIPMF is 52.1% less than KAR-IF, 50% less than PARRA, 47.7% less than ECoRec, 45.2% less than

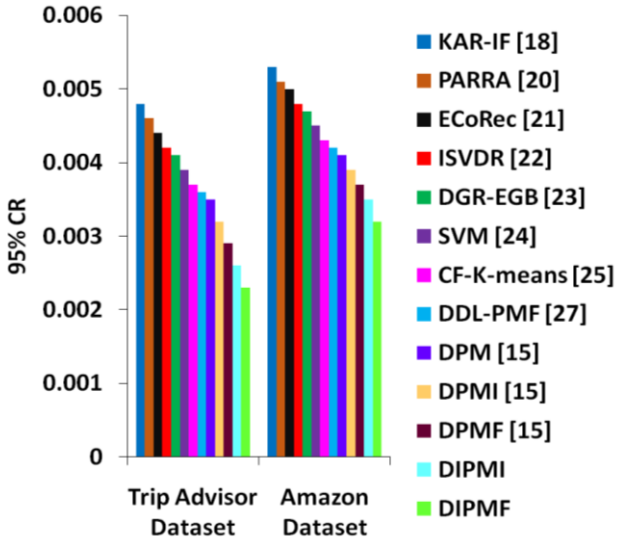

Figure. 7 95% vs. different datasets

ISVDR, 43.9% less than DGR-EGB, 41% less than the SVM, 37.8% less than the CF-K-means, 36.1% less than DDL-PMF, 34.3% less than the DPM, 28.1% less than the DPMI, 20.7% less than the DPMF and 11.5% less than the DIPMI frameworks. Similarly, for the Amazon dataset, the 95% CR of DIPMF is 39.6% less than KAR-IF, 37.3% less than PARRA, 36% less than ECoRec, 33.3% less than ISVDR, 31.9% less than DGR-EGB, 28.9% less than the SVM, 25.6% less than the CF-K-means, 12.8% less than the DDL-PMF, 22% less than the DPM, 18% less than the DPMI, 13.5% less than the DPMF and 8.6% less than the DIPMI frameworks.

Thus, it is concluded that the DIPMF framework achieves higher prediction accuracy and recommendation quality compared to the other big-data recommendation systems efficiently by considering users' preferences, opinions and social contexts for each item to represent the rating behaviors.

5. Conclusion

In this article, a DIPMF framework is suggested for improving the recommendation quality by combining the factors of social context and the dynamic characteristic of each user with their preferences and opinions for each item. This DIPMF has the idea to merge the information from the preferences, opinions and social context of each user. Primarily, the training set is split into the optimum amount of splits to speed up the parallel and distributed prediction. After that, the training task is executed via the distributed predictive MFs with DIPMI for estimating the representation of each user's preferences, opinions and social contexts in the training set. Then, the prediction of rating is formulated as s regression challenge and solved via RF to predict every client's rating behavior with their opinions and social context for every product. To end, the experimental results proved that the DIPMF on the trip advisor dataset has a mean RMSE of 0.6826, mean MAE of 0.5925, quality of 0.8369 and 95% CR of 0.0023 compared to the other frameworks. Similarly, the DIPMF on the Amazon dataset has a mean RMSE of 0.7591, mean MAE of 0.5704, quality of 0.8298 and 95% CR of 0.0032 compared to the other frameworks.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Subbaiah Shanmugasundaram; Methodology, Kavitha Muruganantham; Software, Simulation, Kavitha Muruganantham; Writing- Original draft preparation, Kavitha Muruganantham; Visualization, Investigation, Supervision, Subbaiah Shanmugasundaram; Reviewing and Editing, Subbaiah Shanmugasundaram.

References

[1] A. V. Dev and A. Mohan, "Recommendation system for big data applications based on set similarity of user preferences", In: *Proc. of IEEE International Conf. on Next Generation Intelligent Systems*, pp. 1-6, 2016.

[2] R. Hu, W. Dou, and J. Liu, "ClubCF: a clustering-based collaborative filtering approach for big data application", *IEEE Transactions on Emerging Topics in Computing*, Vol. 2, No. 3, pp. 302-313, 2014.

[3] J. P. Verma, B. Patel, and A. Patel, "Big data analysis: recommendation system with Hadoop framework", In: *Proc. of IEEE International Conf. on Computational Intelligence & Communication Technology*, pp. 92-97, 2015.

[4] Z. Sun, B. Wu, Y. Wu, and Y. Ye, "APL: adversarial pairwise learning recommender systems", *Expert Systems with Applications*, Vol. 118, pp. 573-584, 2019.

[5] R. Wang, H. K. Cheng, Y. Jiang, and J. Lou, "A novel matrix factorization model for recommendation with LOD-based semantic similarity measure", *Expert Systems with Applications*, Vol. 123, pp. 70-81, 2019.

[6] A. M. Turk and A. Bilge, "Robustness analysis of multi-criteria collaborative filtering

algorithms against shilling attacks", *Expert Systems with Applications*, Vol. 115, pp. 386-402, 2019.

[7] R. Zhang, Q. D. Liu, and J. X. Wei, "Collaborative filtering for recommender systems", In: *Proc. of Second IEEE International Conf. on Advanced Cloud and Big Data*, pp. 301-308, 2014.

[8] G. Xu, Z. Tang, C. Ma, Y. Liu, and M. Daneshmand, "A collaborative filtering recommendation algorithm based on user confidence and time context", *Journal of Electrical and Computer Engineering*, Vol. 2019, pp. 1-12, 2019.

[9] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", In: *Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 43-52, 1998.

[10] F. Zhang, T. Gong, V. E. Lee, G. Zhao, C. Rong, and G. Qu, "Fast algorithms to evaluate collaborative filtering recommender systems", *Knowledge-Based Systems*, Vol. 96, pp. 96-103, 2016.

[11] M. R. Zarei and M. R. Moosavi, "A memory-based collaborative filtering recommender system using social ties", In: *Proc. of 4th IEEE International Conf. on Pattern Recognition and Image Analysis*, pp. 263-267, 2019.

[12] S. Ghazarian, and M. A. Nematbakhsh, "Enhancing memory-based collaborative filtering for group recommender systems", *Expert Systems with Applications*, Vol. 42, No. 7, pp. 3801-3812, 2015.

[13] F. Zhang, Y. Lu, J. Chen, S. Liu, and Z. Ling, "Robust collaborative filtering based on non-negative matrix factorization and R1-norm", *Knowledge-Based Systems*, Vol. 118, pp. 177-190, 2017.

[14] W. Serrano, "Intelligent recommender system for big data applications based on the random neural network", *Big Data and Cognitive Computing*, Vol. 3, No. 1, pp. 1-29, 2019.

[15] B. A. Hammou, A. A. Lahcen, and S. Mouline, "An effective distributed predictive model with matrix factorization and random forest for big data recommendation systems", *Expert Systems with Applications*, Vol. 137, pp. 253-265, 2019.

[16] C. Li and K. He, "CBMR: an optimized MapReduce for item-based collaborative filtering recommendation algorithm with empirical analysis", *Concurrency and Computation: Practice and Experience*, Vol. 29, No. 10, pp. 1-7, 2017.

[17] B. A. Hammou, A. A. Lahcen, and S. Mouline, "APRA: an approximate parallel recommendation algorithm for big data", *Knowledge-Based Systems*, Vol. 157, pp. 10-19, 2018.

[18] M. Y. Hsieh, T. H. Weng, and K. C. Li, "A keyword-aware recommender system using implicit feedback on Hadoop", *Journal of Parallel and Distributed Computing*, Vol. 116, pp. 63-73, 2018.

[19] Y. W. Zhang, Y. Y. Zhou, F. T. Wang, Z. Sun, and Q. He, "Service recommendation based on quotient space granularity analysis and covering algorithm on spark", *Knowledge-Based Systems*, Vol. 147, pp. 25-35, 2018.

[20] T. Osadchiy, I. Poliakov, P. Olivier, M. Rowland, and E. Foster, "Recommender system based on pairwise association rules", *Expert Systems with Applications*, Vol. 115, pp. 535-542, 2019.

[21] A. F. D. Costa, M. G. Manzato, and R. J. Campello, "Boosting collaborative filtering with an ensemble of co-trained recommenders", *Expert Systems with Applications*, Vol. 115, pp. 427-441, 2019.

[22] X. Yuan, L. Han, S. Qian, G. Xu, and H. Yan, "Singular value decomposition based recommendation using imputed data", *Knowledge-Based Systems*, Vol. 163, pp. 485-494, 2019.

[23] B. A. Hammou, A. A. Lahcen, and S. Mouline, "A distributed group recommendation system based on extreme gradient boosting and big data technologies", *Applied Intelligence*, Vol. 49, No. 12, pp. 4128-4149, 2019.

[24] G. Yatnalkar, H. S. Narman, and H. Malik, "An enhanced ride sharing model based on human characteristics and machine learning recommender system", *Procedia Computer Science*, Vol. 170, pp. 626-633, 2020.

[25] Q. Shambour, "A deep learning based algorithm for multi-criteria recommender systems", *Knowledge-Based Systems*, Vol. 211, pp. 1-15, 2021.

[26] A. F. O. U. D. I. Yassine, L. A. Z. A. A. R. Mohamed, and M. A. Achhab, "Intelligent recommender system based on unsupervised machine learning and demographic attributes", *Simulation Modelling Practice and Theory*, Vol. 107, pp. 1-17, 2021.

[27] Hanafi, E. Pujastuti, A. Laksito, R. Hardi, R. Perwira, A. Arfriandi, and Asroni, "Handling sparse rating matrix for e-commerce recommender system using hybrid deep learning based on LSTM, SDAE and latent

factor", *International Journal of Intelligent Engineering & Systems*, Vol. 15, No. 2, pp. 379-393, 2022, doi: 10.22266/ijies2022.0430.35.