

Received April 25, 2020, accepted May 26, 2020, date of publication June 1, 2020, date of current version June 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2998824

A Bayesian Inference Based Hybrid Recommender System

ARMIELLE NOULAPEU NGAFFO^{ID}, (Graduate Student Member, IEEE),
WALID EL AYEB, AND ZIÈD CHOUKAIR

Mediatron Laboratory, Higher School of Communications of Tunis (SUP'COM), University of Carthage, Carthage 1054, Tunisia

Corresponding author: Armielle Noulapeu Ngaffo (armielle.ngaffo@supcom.tn)

This work was supported by the Central Africa Backbone (CAB) Project in Cameroon under Grant P-CM-GB0-002.

ABSTRACT The large mass of various products/services accessible on the Internet has motivated the development of recommender systems to refine the selection of items aligned with users' expectations. Recommender systems have been developed to tackle the item targeting problem. They are crucial tools that quickly target items fitting users' needs, thus allowing them to easily identify the items that fit their tastes and preferences. Following state-of-the-art methods, a distinction is made between content-based recommender approaches and collaborative filtering-based recommender approaches. Collaborative filtering-based recommender approaches are the most widely adopted methods. They are divided into memory-based methods that show the advantage of their easy-understandability, and model-based methods that are data sparsity resilient and high-accurate. In this paper, we propose a hybrid model-based recommendation approach, a combination of a user-based approach and an item-based approach. Our method estimates the probability with which a user would rate an item. It performs a Bayesian inference of future end-user interests and shows the advantage of the easy-understandability of memory-based methods and the effectiveness of model-based methods. Experiments are conducted on real-world datasets and show that our method outperforms several state-of-the-art recommendation methods regarding the prediction accuracy and the recommendation quality.

INDEX TERMS Bayesian inference, recommender system, Dirichlet distribution, collaborative filtering, maximum a posteriori estimation.

I. INTRODUCTION

The rapid growth of the Internet has boosted the increase of rich and varied content (videos, films, music, articles, services etc.). However, the diversity and the large number of items accessible by users complexify the selection of items aligned with users' needs [1], [2]. To address the information overload problem, Recommender Systems (RS) have been developed to refine the items' targeting. Therefore, RS are decision support tools for end-users. According to the literature, RS are organized into content-based methods and Collaborative Filtering (CF)-based methods [3]. Content-based methods rely on the past users' experiences in order to predict their future interests. They are strongly focused on the target user's profile, and therefore they are not effective in case of poor user profile [4]. To overcome content-based methods limitations, CF-based approaches lay on profiles of a group of users who share the same or similar tastes. CF-based

methods are built on the assumption that within a group of users with preferences identical or similar to those of a target user, the historical data of some of them can be used to predict the future interests of the target user [5]. CF-based approaches are very popular and organized into memory-based CF and model-based CF [6]. From a users' rating matrix, memory-based methods rely on computations of inter-user similarities according to user-based CFs, or inter-item similarities regarding item-based CFs [7]. They are easy-explainable and simple-implementable but show poor performances in case of users' data sparseness [8]. Meanwhile, model-based methods are complex but show a high precision even for large datasets [9]. They alleviate the data sparsity problem and the cold start problem [10]. Indeed, they are effective even when users' data are insufficient or when a new user or a new item is added to the system and no relevant information are available to improve the related recommendation.

The matrix factorization technique is widely adopted for the implementation of model-based CFs [11]. This technique performs a reduction of the original data matrix.

The associate editor coordinating the review of this manuscript and approving it for publication was Tomás F. Peña^{ID}.

Following the matrix factorization technique, the prediction of the target user's interests is carried out on a reduced data matrix. Consequently, relevant data are lost during the latent feature modeling that is a hard masterable process [7]. The data lost during the matrix factorization process affects the prediction accuracy and therefore the item targeting precision. To alleviate the item targeting problem, in this paper, we propose a hybrid recommender system that uses Bayesian estimation to predict users' interests. Our proposal combines a user-based approach and an item-based approach to assess the probability with which a user rates an item. In other words, the user's sensitivity relative to a given item is estimated by maximizing the posterior probability with which a user rates an item. Recall that, the user-based approach of our model refers to infer the probability with which a user rates an item knowing that users with similar profiles have rated the item; while the item-based approach of our model refers to estimate the probability with which an item is rated by a user knowing that similar items have been rated by the user.

Our contribution in this paper is highlighted by the proposal of a hybrid Bayesian recommender model showing the advantage of the easy-explainability of memory-based CFs and the effectiveness of model-based CFs. Indeed, although high-effective, the hard-understandability of model-based CFs complexifies their implementation. Our method is easy-explainable and high-accurate. In this paper, our contributions are declined as follows:

- A novel Bayesian inference-based model (BIHRS) is proposed to predict users' interests. Our model uses a Dirichlet distribution to accurately model a priori information such as the item's popularity, the usual behavior adopted by a large number of users, the consumption habits, and preferences shared by a majority of users.
- An easy-explainable and high-accurate recommendation approach is developed by combining assets of memory-based CFs and model-based CFs. Indeed, the proposed recommender approach benefits from a hybridization process that enables the inclusion of all various aspects raised by interactions between users and items; and therefore increases the resilience of our method to the data sparsity problem. With an increased data-sparseness resilience thanks to an effective hybridization, our method shows a valuable recommendation accuracy.
- Extensive experimentations are conducted on various real-world datasets to highlight the effectiveness and the scalability of our proposal.

The remainder of this paper is organized as follows: section II surveys state-of-the-art recommendation methods, section III details our recommendation approach. Section IV describes the conducted experiments and obtained results. Section V concludes this paper and presents our perspectives.

II. RELATED WORK

The information overload on the Internet has complexified the selection of items relevant regarding users' needs.

The challenges in the RS's field have motivated researchers to develop recommendation methods. In this section, we survey state-of-the-art methods employed in CF-based approaches as CFs are part of the most influential recommendation algorithms [8] and are related to our proposal.

A. MEMORY-BASED METHODS

Memory-based CFs exploit users' ratings relative to items to make the recommendation. For this purpose, a user-item matrix is built from the collected ratings. Afterwards, inter-user similarities are computed to assess the proximity of tastes between users [7]. Finally, the recommendation is based on a weighted average of ratings of users having similar preferences [8]. Another variant of memory-based CF focuses on items and computes inter-item similarities to assess the proximity of users' interests raised by items [12].

In [13], the authors propose a recommender system based on inter-user trust relationships. They define a measure of inter-user similarity from a customized Pearson Correlation Coefficient (PCC) and the Jaccard Correlation Coefficient (JCC) [14]. Subsequently, the inter-user trustworthiness is assessed in order to guarantee the recommendation reliability.

The authors in [15] propose a cloud service recommendation approach based on the evaluation of inter-user similarity using the Spearman Rank Correlation Coefficient (SRCC). The authors' proposal presents the advantage of ranking prediction accuracy. In [16], the authors propose a similar recommendation approach but they apply the data smoothing technique to improve the recommendation quality. Following [17], the authors propose a recommendation model based on a customized K-means algorithm combined with the cuckoo search to aggregate users with similar profiles. Regarding [18], the authors propose a measure of inter-user similarity based on Bayes concordance rate to improve the recommendation precision. In [12], the authors propose an efficient privacy-preserving recommendation approach. The authors' method is based on an item-based CF. To preserve user privacy, the authors assess inter-item similarity using a customized PCC. Memory-based CFs used in the aforementioned researches present the advantage of being easy-understandable and easy-implementable. However, they are unsuitable for large datasets because they are high-computational cost methods, and therefore non-scalable and low-accurate. Indeed, the similarity computations are performed on a full high-dimensional rating matrix inducing an increase of the computational cost. In addition, referring to the data sparsity problem, these proposals show poor performances when a large amount of relevant data is missing.

B. MODEL-BASED METHODS

Model-based CFs apply machine learning techniques to build from users' data, a model that better fits observations [19]. Model-based methods are high-accurate and data sparsity-resilient. It means that they are effective even when facing the missingness of significant information relevant to the recommendation refinement [20].

The matrix factorization technique is widely used in model-based CFs due to its efficiency in modeling users' preferences from a small number of latent factors [21]. This technique consists in reducing high-dimensional users' data matrix into low-dimensional matrices thanks to an accurate selection of relevant latent features [22]. Thereafter, an optimization method is applied to minimize the error due to the matrix reduction [11].

Assuming that users' data can be modeled by low-dimensional latent factors matrices, in [23], the authors use the matrix factorization technique to improve the recommendation accuracy. To tackle the data sparsity problem and the cold-start problem, in [24], the authors propose a probabilistic matrix factorization framework for an online recommendation. The authors in [25], present a non-negative variant of matrix factorization that integrates social trust information in a model that addresses data sparsity and cold start issues. With a similar trend, the authors in [26] integrate social information into their recommender system based on a matrix factorization method. From implicit feedbacks, the authors in [27] propose a personalized ranking method using a Bayesian pairwise learning to improve the recommendation performances that are based on the matrix factorization technique. The authors in [28], propose a time-aware recommendation approach that uses a matrix factorization technique and a Long Short Term Memory (LSTM) model for the selection of services from the Quality of Service (QoS) prediction. In [29], [30], authors propose a matrix factorization based method that uses a Bayesian strategy. In [29], they propose a probabilistic matrix factorization based recommendation approach that they extend in [30] by applying the Gibbs sampling strategy that speeds up the algorithm convergence and therefore enhances the execution time of their method.

To alleviate the data sparseness issue, the authors in [31] propose a hybrid recommendation approach for tourist spots. Their proposal assesses the correlation between image features using a Bayesian ranking algorithm. In [32], the authors propose a model-based CF using a parallel naive Bayesian approach implemented on a Hadoop distributed infrastructure. While researchers in [33] infer the potential risk of disease using Bayesian collaborative filtering. In [34], the authors develop an electricity plan recommender system using a probabilistic approach to predict the missing features used in the recommendation of electricity retailing plans. The authors' approach is based on a matrix factorization technique to reduce the computational cost of their algorithm. The authors in [35], propose a recommendation approach based on a naive Bayesian classifier to predict the probability with which a user rates an item. However, the authors use a uniform probability distribution that does not consider a priori intuitions related to the item's popularity, the consumption habits adopted by the majority of users. Authors in [36], [37] propose recommendation approaches based on Bayesian networks. Those methods are both based on the Naive Bayesian classifier that benefits from the effectiveness of model-based approaches. In [37], they use Laplace estimator in the

prediction process, while authors in [36] combine content-based and CF-based approaches to enhance the prediction accuracy. However, those methods both employ uniform a priori probabilities that do not hold some real considerations. Indeed, in practice users rate only a small subset of items. In this way, some items are more popular than others, and they therefore show a higher a priori probability to be selected by users compared to other items. It means that a uniform a priori probability will not consider those aspects. Furthermore, users who rate most of the items show a higher a priori probability to select an item. For this purpose, our proposal lays on non-uniform probability distributions and hybridization between a user-based approach and an item-based approach in order to consider all aspects raised by interactions between users and items, and therefore enhance the prediction accuracy. In addition, the proposed method gains accuracy by considering non-uniform probabilities that consider the aforementioned aspects of user-item interactions.

Based on the matrix factorization technique, the above-surveyed researches present the advantage of being high accurate and efficient for large datasets. However, the matrix factorization technique is mainly based on the decomposition into low-dimensional latent factor matrices which is a complex, hard-explainable and hard-masterable process. Meanwhile the above-presented bayesian researches are based on uniform probability distributions that do not model the a priori information in a realistic way given the observed data.

C. HYBRID RECOMMENDATION METHODS

Hybrid recommender systems have been developed to alleviate limitations shown by existing user-based CF, item-based CF, content-based, and model-based approaches separately [38]. In hybrid approaches, predictions are performed on each method involved in the hybridization process separately, and thereafter combined to overcome some common problems such as cold start problem and data sparseness [39]. In the literature, several approaches [40]–[42] are developed to benefit from hybrid recommender systems. The main advantage of those methods is their ability to employ plural aspects raised by interactions between users and items to refine as most as possible the recommendation. In [43], the authors propose a hybrid recommendation approach that combines several explanations styles (user-based, item-based, popularity-based) to refine the music recommendations. Authors in [44] propose a recommendation method that merges item-based CF, user-based CF, and factor-based approaches in order to build a hybrid recommender system for artists. Researchers in [45] exploit user-item interactions and data from purchased items to propose a hybrid system for artist recommendation. As the above-presented methods, existing hybrid recommender systems are based on a combination of recommendation approaches which have shown some limitations. Their effectiveness, therefore, remains limited by the drawbacks of methods involved in the hybridization process. To remedy this, the hybridization of our proposal lays on Bayesian user-based and item-based approaches that

are separately efficient before the effectiveness of the hybrid method.

In this paper, we embed the hybridization concept to benefit from various aspects raised by user-items interactions. We perform a hybrid prediction as each user presents an interest in specific items; and reversely, each item raises an interest observed from a set of users who have consumed it. To address the fast items identification problem, we propose an easy-explainable and high-accurate Bayesian recommendation approach. Driven by Dirichlet distributions, our proposal considers a priori information and users' intuitions to accurately model the observed ratings.

The next section presents our BIHRS approach.

III. BIHRS RECOMMENDATION METHOD

The proposed recommendation approach is flexible and configurable since any kind of actions (clicks, likes, purchases, reviews ...) that users perform on items can be transformed into scores assimilable to ratings. In this way, the degree of the users' interest on items is therefore scored like an explicit rating would do. This configurability of our model contributes to the robustness of the system to the cold-start problem that happens when a new item or a new user is added to the system knowing that there are no explicit ratings on the new item or from the new user. In the rest of the paper, our model is fed by ratings assigned by users on items knowing that any other type of users' actions on items can be scored as explicit ratings and involved to feed our prediction model.

Our approach aims to estimate the probability with which a user u would rate an item i with an unknown rating \tilde{r}_{ui} . Recall that each rating assigned to an item by a user reflects the level of interest expressed by the user relatively to the concerned item. In other words, to predict the future interest of a user u , it is required to infer the probability with which he would rate an item i . The Bayesian approach used in our model first identifies the probability distribution that better fits the observed ratings. Thereafter, the probability model of the estimation parameters is built in order to generalize the probability distribution of observations and ratings likely to be assigned on item in the future by the active user. Following the Bayesian approach, Fig. 1 describes the workflow of our BIHRS estimation model as follows:

- The historical users' experiences regarding items are expressed via a D tensor of users' ratings probabilities, and via a B tensor of probabilities of ratings on items. The D tensor is fed by the dataset of users' ratings while the B tensor is fed by the dataset of observations on items.
- Subsequently, we estimate the $h_u(\Omega)$ prior probability distribution of the users' cluster membership and the overall rating probability of each users' cluster. Analogically, regarding the item-based approach, we calculate the $h_i(\Lambda)$ prior probability distribution of the items' cluster membership and the overall rating probability by each items' cluster.

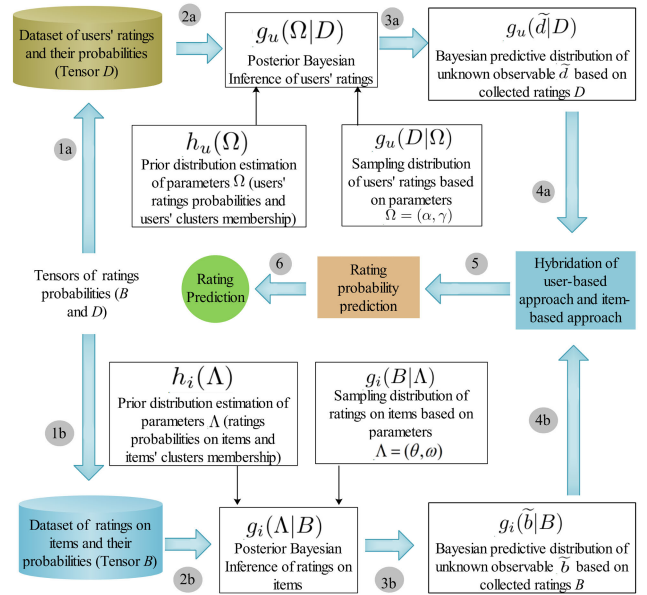


FIGURE 1. The BIHRS Model.

- We evaluate the $g_u(D|\Omega)$ sampling distribution of users' ratings considering the α users' cluster membership parameter and the γ overall rating probability parameter of each users' cluster. From the item-based approach, we compute the $g_i(B|\Lambda)$ sampling distribution of ratings observed on items considering the θ items' cluster membership and the ω overall rating probability of each items' cluster.
- Based on the $h_u(\Omega)$ prior probability distribution, we infer the $g_u(\Omega|D)$ posterior probability distribution of users' ratings considering D observations. Thereafter, we infer the $g_u(\tilde{d}|D)$ Bayesian predictive distribution of unknown observable \tilde{d} given the D users' observations. Considering the D users' observations, the $g_u(\tilde{d}|D)$ distribution evaluates the \tilde{d} probability with which a user u would rate an item i by assigning it an unknown rating \tilde{r}_{ui} . From the item-based approach, based on the $h_i(\Lambda)$ prior probability distribution, we estimate the $g_i(\tilde{b}|B)$ Bayesian predictive distribution of unknown observable \tilde{b} considering the B observations on items. Given the B observations on items, the $g_i(\tilde{b}|B)$ distribution prospectively evaluates the \tilde{b} unknown probability with which an item i is rated by a user u .
- From Bayesian predictive distributions $g_u(\tilde{d}|D)$ and $g_i(\tilde{b}|B)$, the hybridization is performed to obtain a hybrid Bayesian predictive distribution \tilde{p}_{ui} . Following the Maximum a Posteriori (MAP) principle, the unknown rating \tilde{r}_{ui} is deduced by maximizing \tilde{p}_{ui} .

To increase the robustness of our model related to data sparsity, we build clusters using the popular K-means algorithm [46]. To enhance the clustering performance, we elect centroids of clusters following a specific scheme. At the initialization of the clustering process, the first centroid is

the user who has rated the highest number of items to get more common items between him and other users. The other centroids are thereafter selected among users in such a way that maximizes the distance between them and existing centroids. That distance is characterized by a probability defined as follows [47]:

$$p = \frac{Dist(u')^2}{\sum_{u \in U} Dist(u)^2}, \tag{1}$$

where u' is a potential centroid. and $Dist$ is computed as follows:

$$Dist(u') = \frac{1}{Sim_{PCC} + \tau}, \tag{2}$$

where Sim_{PCC} is the similarity evaluated between u' and existing centroids according to the Pearson Correlation Coefficient (PCC) [8], and $\tau \geq 1$ is a positive value that avoids negative distance due to values of PCC that range between -1 and 1 . Through an analogical approach, we initialize items clustering process by selecting as the first centroid, the item that has been most often rated. Other centroids are selected among items the most distant from existing centroids. Once elected centroids, the remaining of the process consists to populate clusters by adding users and items that are the most similar to centroids.

Our estimation model is based on the distribution of all users into users' clusters. In accordance with the principle of collaborative filtering, the idea is to use the historical experiences of certain users of the users' cluster to infer the other users' future experiences. Each users' cluster aggregates users with similar preferences; to mean users who have historically assigned similar ratings to the same items. From a probabilistic approach, considering users' profiles, a user belongs to a users' cluster CLU_t with an a priori probability α_t . By similar reasoning, items are also organized into items' clusters. An item is assigned to an items' cluster according to the proximity of ratings on it. An item belongs to an items' cluster CLI_l with an a priori probability θ_l . Fig. 2 describes the probability distribution of users and items clusters membership.

The next subsection presents our mathematical model.

A. PROBLEM FORMULATION

We consider the set of M users $U = \{u_1, u_2, \dots, u_M\}$, and the set of N items $I = \{i_1, i_2, \dots, i_N\}$. Users are divided into t clusters, $t \in T$ where $T = \{1, 2, \dots, C\}$ is the set of the number of users' clusters. Items are organized in l clusters, $l \in L$ where $L = \{1, 2, \dots, H\}$ is the set of the number of items' clusters. Users rate items by assigning them r values; $r \in R = \{r_1, r_2, \dots, r_S\}$.

We define $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_t, \dots, \alpha_C\}$, the vector of probabilities of membership to the respective users' clusters $CLU_1, CLU_2, \dots, CLU_t, \dots, CLU_C$. Each user belongs to a cluster CLU_t with a probability α_t , where $t \in T$.

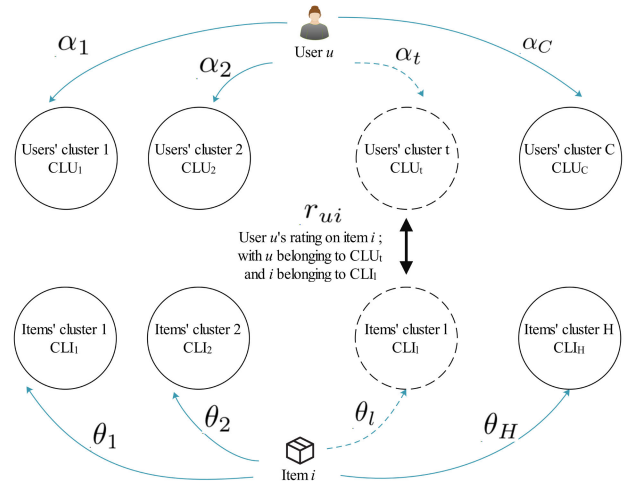


FIGURE 2. User and Item Clusters Membership Probabilities.

We consider $\theta = \{\theta_1, \theta_2, \dots, \theta_l, \dots, \theta_H\}$, the vector of probabilities of membership to the respective items' clusters $CLI_1, CLI_2, \dots, CLI_l, \dots, CLI_H$. Each item belongs to a cluster CLI_l with a probability θ_l , where $l \in L$.

B. USER-BASED APPROACH

We consider the tensor $D \in \Psi^{M \times N \times S}$, where $\Psi \in [0, 1]$. The tensor $D = [d_{uir}]_{M \times N \times S}$ is defined by using the U set of users, the I set of items, and the R set of ratings. This tensor hosts the d_{uir} probabilities of the rating r assigned by a user u on an item i . We consider $\gamma = \{\gamma_1, \gamma_2, \dots, \gamma_t, \dots, \gamma_C\}$, the vector of users' ratings probabilities belonging to clusters $CLU_1, CLU_2, \dots, CLU_t, \dots, CLU_C$ respectively. According to the Bayesian theory [48], we set the estimation parameter $\Omega = (\alpha, \gamma)$ made from probabilities of users' cluster membership and users' rating probabilities by cluster.

The purpose is to predict the probability with which a user u rates an item i given the Ω estimation parameter and the D set of historical observations known as past observed users' ratings. Following the Bayesian inference, we estimate the $g_u(\tilde{d}|D)$ predictive distribution of a \tilde{d} unknown observable given the D set of observed users' ratings. For this purpose, we implement in Algorithm 1 a progressive reasoning declined as follows:

- Estimation of the $g_u(D|\Omega)$ sampling distribution also known as the Likelihood distribution according to Bayesian theory [49]. This distribution is defined as the probability distribution of all users' ratings.
- Estimation of the $h_u(\Omega)$ prior distribution defined as the probability distribution of Ω estimation parameter.
- Estimation of the $g_u(\Omega|D)$ posterior distribution known as the probability distribution of the Ω parameter based on observations.
- Estimation of the $g_u(\tilde{d}|D)$ Bayesian predictive distribution defined as the probability distribution for an unknown observable \tilde{d} .

Considering a user u belonging to the cluster CLU_t with a α_t cluster membership probability. His $g_u(d_u|\Omega)$ rating probability distribution is trivially described as a multinomial distribution defined as follows:

$$g_u(d_u|\Omega) = \alpha_t \prod_{i \in I} \prod_{r \in R} d_{uir}^{v_{uir}}, \quad (3)$$

where v_{uir} is a random variable set to 1 when user u rates an item i with rating r , and 0 otherwise; d_u is the known observable relative to user u ; and d_{uir} is the probability with which the active user u rates an item i .

Given the fact that the user u can belong to any cluster $CLU_t, t \in T$, the $g_u(d_u|\Omega)$ is reexpressed as follows:

$$g_u(d_u|\Omega) = \sum_{t=1}^C \alpha_t \prod_{i \in I} \prod_{r \in R} d_{uir}^{v_{uir}}. \quad (4)$$

The overall $g_u(D|\Omega)$ users' rating probability distribution [50] also known as the sampling distribution or Likelihood distribution, is described as follows:

$$g_u(D|\Omega) = \prod_{u \in U} \sum_{t=1}^C \alpha_t \prod_{i \in I} \prod_{r \in R} d_{uir}^{v_{uir}}. \quad (5)$$

The next step is to estimate the $h_u(\Omega)$ prior distribution of the Ω estimation parameter. The $h_u(\Omega)$ prior probability distribution aims to model a priori user-related information such as the most common users' consumption habits. To estimate $h_u(\Omega)$, it is mathematically convenient to choose a conjugate distribution to reduce the estimation process complexity [48]. The Dirichlet distribution is the right conjugate prior distribution given our $g_u(D|\Omega)$ Likelihood distribution that is a multinomial probability function [51].

Because α and γ parameters are both independent vectors, the $h_u(\Omega)$ prior distribution is deductively computed as follows:

$$h_u(\Omega) = h_u(\alpha).h_u(\gamma), \quad (6)$$

where $h_u(\alpha)$ and $h_u(\gamma)$ are partial prior distributions defined by Dirichlet functions.

The $h_u(\alpha)$ distribution is expressed as follows:

$$\begin{aligned} h_u(\alpha) &= \frac{\Gamma(\lambda_1 + \dots + \lambda_K)}{\Gamma(\lambda_1) + \dots + \Gamma(\lambda_K)} \prod_{j=1}^K \prod_{t=1}^C \alpha_t^{\lambda_j-1} \\ &= \frac{\Gamma(\sum_{j=1}^K \lambda_j)}{\prod_{j=1}^K \Gamma(\lambda_j)} \prod_{j=1}^K \prod_{t=1}^C \alpha_t^{\lambda_j-1} \\ &= \frac{1}{Be(\lambda)} \prod_{j=1}^K \prod_{t=1}^C \alpha_t^{\lambda_j-1}, \end{aligned} \quad (7)$$

where $\lambda_1, \lambda_2, \dots, \lambda_K$ are Beta function hyperparameters. For a non-informative prior [48], we set $K = 1$ and $\lambda = 1$.

The $h_u(\gamma)$ partial prior distribution is computed as follows:

$$\begin{aligned} h_u(\gamma) &= \frac{\Gamma(\mu_1 + \dots + \mu_K)}{\Gamma(\mu_1) + \dots + \Gamma(\mu_K)} \prod_{j=1}^K \prod_{r \in R} d_{uir}^{\mu_j-1} \\ &= \frac{\Gamma(\sum_{j=1}^K \mu_j)}{\prod_{j=1}^K \Gamma(\mu_j)} \prod_{j=1}^K \prod_{r \in R} d_{uir}^{\mu_j-1} \\ &= \frac{1}{Be(\mu)} \prod_{j=1}^K \prod_{r \in R} d_{uir}^{\mu_j-1}, \end{aligned} \quad (8)$$

where $\mu_1, \mu_2, \dots, \mu_K$ are Beta function hyperparameters. As for the $h_u(\alpha)$ distribution, for a non-informative prior, we set $K = 1$ and $\mu = 1$.

After estimating the $g_u(D|\Omega)$ Likelihood and the $h_u(\Omega)$ prior distributions, we infer the $g_u(\tilde{d}|D)$ Bayesian predictive distribution for an unknown observable \tilde{d} . For this purpose, we first estimate the $g_u(\Omega|D)$ overall posterior distribution of Ω estimation parameter given the D set of observed users' ratings. Thereafter, from the $g_u(\Omega|D)$ posterior distribution, the $g_u(\tilde{d}|D)$ Bayesian predictive distribution is deduced. According to the Bayesian theory [49], the $g_u(\Omega|D)$ posterior distribution is computed as follows:

$$\begin{aligned} g_u(\Omega|D) &\propto h_u(\Omega).g_u(D|\Omega) \\ &\propto h_u(\alpha).h_u(\gamma).g_u(D|\Omega), \end{aligned} \quad (9)$$

where $h_u(\alpha)$ and $h_u(\gamma)$ are prior distributions; and $g_u(D|\Omega)$ is the Likelihood distribution computed in Equation (5).

According to [48], the $g_u(\tilde{d}|D)$ Bayesian predictive distribution is computed as follows:

$$\begin{aligned} g_u(\tilde{d}|D) &= \int_{\Omega} g_u(\tilde{d}, \Omega|D)d(\Omega) \\ &= \int_{\Omega} g_u(\tilde{d}|\Omega, D)g_u(\Omega|D)d(\Omega), \end{aligned} \quad (10)$$

where $g_u(\Omega|D)$ is the overall posterior distribution of the estimation parameter given the observed users' ratings; and $g_u(\tilde{d}|\Omega, D)$ is the joint sampling distribution for the unknown observable \tilde{d} given the D set of observations.

The integral in the $g_u(\tilde{d}|D)$ expression is complex to compute. To reduce the computational complexity, we use a numerical integration for a straight computation [49]. The $g_u(\tilde{d}|D)$ distribution is reexpressed as follows:

$$g_u(\tilde{d}|D) = \sum_{t=1}^C g_u(\tilde{d}|\alpha_t, D).g_u(\alpha_t|D). \quad (11)$$

The $g_u(\tilde{d}|\alpha_t, D)$ distribution component of Equation (11) is estimated as follows [48]:

$$\begin{aligned} g_u(\tilde{d}|\alpha_t, D) &\propto \int_{\gamma} g_u(\tilde{d}|\alpha_t, \gamma)g_u(D|\alpha_t, \gamma)h_u(\gamma)d\gamma \\ &\propto \int_{\gamma} \frac{\alpha_t}{Be(\mu)} \prod_{r \in R} d_{uir}^{v_{ur} + \mu - 1 + \sum_{j \in CLU_t} v_{jir}} d\gamma. \end{aligned} \quad (12)$$

The integral in Equation (12) can be approximated by a Beta function as follows:

$$g_u(\tilde{d}|\alpha_t, D) \propto \frac{\alpha_t}{Be(\mu)} * \underbrace{Be(v_{ui} + \mu + \sum_{j \in CLU_t} v_{jir}, \dots, v_{ui} + \mu + \sum_{j \in CLU_t} v_{jir})}_{R\text{-dimension vector}} \tag{13}$$

From the original definition of the Beta function through the Gamma function, Equation (13) can be rewritten as follows:

$$g_u(\tilde{d}|\alpha_t, D) \propto \frac{\alpha_t}{Be(\mu)} \frac{\prod_{r \in R} \Gamma(\mu + \sum_{j \in CLU_t} v_{jir})}{\Gamma(\sum_{r \in R} (\mu + \sum_{j \in CLU_t} v_{jir}))} \tag{14}$$

By averaging Equation (14), the $g_u(\tilde{d}|\alpha_t, D)$ distribution is approximated as follows:

$$g_u(\tilde{d}|\alpha_t, D) \propto \frac{\alpha_t}{Be(\mu)} \frac{\mu + \sum_{j \in CLU_t} v_{jir}}{R\mu + \sum_{r \in R} \sum_{j \in CLU_t} v_{jir}} \tag{15}$$

According to the Bayesian theory, the $g_u(\alpha_t|D)$ distribution component of Equation (11) is estimated as follows:

$$g_u(\alpha_t|D) \propto \int_{\Omega} g_u(D|\alpha_t, \gamma) h_u(\alpha) h_u(\gamma) d\Omega \propto \int_{\Omega} \left(\prod_{i \in I} \prod_{r \in R} \alpha_t d_{uir}^{\sum_{j \in CLU_t} v_{jir}} \right) * \left(\frac{1}{Be(\lambda)} \prod_{t=1}^C \alpha_t^{\lambda-1} \right) \left(\frac{1}{Be(\mu)} \prod_{r \in R} d_{uir}^{\mu-1} \right) d\Omega \propto \frac{1}{Be(\lambda)Be(\mu)} \int_{\Omega} \left(\prod_{t=1}^C \alpha_t^{\lambda} \right) * \left(\prod_{i \in I} \prod_{r \in R} d_{uir}^{\mu-1 + \sum_{j \in CLU_t} v_{jir}} \right) d\Omega, \tag{16}$$

The integral in Equation (16) can be approximated by a Beta function as follows:

$$g_u(\alpha_t|D) \propto \frac{1}{Be(\lambda)Be(\mu)} \underbrace{Be(\lambda + 1, \dots, \lambda + 1)}_{C\text{-dimension vector}} * \prod_{i \in I} \underbrace{Be(\mu + \sum_{j \in CLU_t} v_{jir}, \dots, \mu + \sum_{j \in CLU_t} v_{jir})}_{R\text{-dimension vector}}$$

Algorithm 1 User-Based BIHRS Algorithm

Data:

- D : Tensor of users' ratings probabilities
- α : Vector of probabilities of users' cluster membership
- γ : Vector of ratings probabilities by users' cluster
- μ, λ : Beta function hyperparameters
- I : Set of items
- U : Set of users
- C : Number of users' clusters

Result: $g_u(\tilde{d}|D)$ the Bayesian predictive distribution for an unknown observable \tilde{d}

1 Begin

2 Compute the sampling distribution

$$g_u(D|\Omega) = \prod_{u \in U} \prod_{t=1}^C \alpha_t \prod_{i \in I} \prod_{r \in R} d_{uir}^{v_{uir}};$$

3 Compute the partial prior distribution $h_u(\alpha)$ using (7);

4 Compute the partial prior distribution $h_u(\gamma)$ using (8);

5 Compute the joint prior distribution

$$h_u(\Omega) = h_u(\alpha).h_u(\gamma);$$

6 Compute the posterior distribution

$$g_u(\Omega|D) = h_u(\Omega).g_u(D|\Omega);$$

7 **for** t from 1 to C **do**

8 $g_u(\tilde{d}|\alpha_t, D) = \frac{\alpha_t}{Be(\mu)} \frac{\mu + \sum_{j \in CLU_t} v_{jir}}{R\mu + \sum_{r \in R} \sum_{j \in CLU_t} v_{jir}};$

9 $g_u(\alpha_t|D) = \frac{1}{Be(\lambda)Be(\mu)} \left(\frac{\lambda+1}{C\lambda+C} \right) \prod_{i \in I} \frac{\mu + \sum_{j \in CLU_t} v_{jir}}{R\mu + \sum_{r \in R} \sum_{j \in CLU_t} v_{jir}};$

10 Compute the Bayesian predictive distribution for an unknown observable \tilde{d}

$$g_u(\tilde{d}|D) = g_u(\tilde{d}|\alpha_t, D).g_u(\alpha_t|D);$$

11 **end**

12 Return $g_u(\tilde{d}|D)$;

$$\propto \frac{1}{Be(\lambda)Be(\mu)} \frac{\prod_{t=1}^C \Gamma(\lambda + 1)}{\Gamma(\sum_{t=1}^C (\lambda + 1))} \prod_{i \in I} \frac{\prod_{r \in R} \Gamma(\mu + \sum_{j \in CLU_t} v_{jir})}{\Gamma(\sum_{r \in R} (\mu + \sum_{j \in CLU_t} v_{jir}))} \propto \frac{1}{Be(\lambda)Be(\mu)} \frac{\prod_{t=1}^C \Gamma(\lambda + 1)}{\Gamma(C\lambda + C)} \prod_{i \in I} \frac{\prod_{r \in R} \Gamma(\mu + \sum_{j \in CLU_t} v_{jir})}{\Gamma(R\mu + \sum_{r \in R} \sum_{j \in CLU_t} v_{jir})}, \tag{17}$$

By averaging Equation (17), the $g_u(\alpha_t|D)$ distribution is estimated as follows:

$$g_u(\alpha_t|D) \propto \frac{1}{Be(\lambda)Be(\mu)} \left(\frac{\lambda+1}{C\lambda+C} \right) \prod_{i \in I} \frac{\mu + \sum_{j \in CLU_t} v_{jir}}{R\mu + \sum_{r \in R} \sum_{j \in CLU_t} v_{jir}}, \tag{18}$$

Algorithm 1 summarizes the Bayesian inference process following the user-based approach.

The next subsection details the Bayesian inference process regarding the item-based approach.

C. ITEM-BASED APPROACH

We consider the tensor $B \in \Phi^{N \times M \times S}$, where $\Phi \in [0, 1]$. The tensor $B = [b_{iur}]_{N \times M \times S}$ is defined by using the I set of items, the U set of users, and the R set of ratings. This tensor hosts the b_{iur} probabilities of a rating r on an item i and assigned by a user u . We consider $\omega = \{\omega_1, \omega_2, \dots, \omega_l, \dots, \omega_H\}$, the vector of ratings probabilities on items belonging to clusters $CLI_1, CLI_2, \dots, CLI_l, \dots, CLI_H$ respectively. We consider $\theta = \{\theta_1, \theta_2, \dots, \theta_l, \dots, \theta_H\}$, the vector of membership probabilities respectively related to the items' clusters $CLI_1, CLI_2, \dots, CLI_l, \dots, CLI_H$. Following the Bayesian theory, we set the estimation parameter $\Lambda = (\theta, \omega)$ from probabilities of items' cluster membership and rating probabilities on items by cluster.

As for the user-based approach, in the item-based approach, we adopt in Algorithm 2 a progressive reasoning declined as follows:

- Estimation of the $g_i(B|\Lambda)$ sampling distribution of all ratings on items given the Λ parameter.
- Estimation of the $h_i(\Lambda)$ prior distribution of Λ estimation parameter.
- Estimation of the $g_i(\Lambda|B)$ posterior distribution that refers to the probability distribution of the Λ parameter based on observations.
- Estimation of the $g_i(\tilde{b}|B)$ Bayesian predictive distribution for an unknown observable \tilde{b} .

Let an item i belonging to the cluster CLI_l with a θ_l cluster membership probability. Its $g_i(b_i|\Lambda)$ rating probability distribution is trivially described as a multinomial distribution defined as follows:

$$g_i(b_i|\Lambda) = \theta_l \prod_{u \in U} \prod_{r \in R} b_{iur}^{w_{iur}}, \tag{19}$$

where w_{iur} represents a random variable set to 1 when item i receives a rating r assigned by user u , and 0 otherwise; b_i is the known observable relative to item i .

Knowing that item i can belong to any cluster $CLI_l, l \in L$, the $g_i(b_i|\Lambda)$ is reexpressed as follows:

$$g_i(b_i|\Lambda) = \sum_{l=1}^H \theta_l \prod_{u \in U} \prod_{r \in R} b_{iur}^{w_{iur}}. \tag{20}$$

The overall $g_i(B|\Lambda)$ rating probability distribution on item i , also known as the Likelihood or sampling distribution, is described as follows:

$$g_i(B|\Lambda) = \prod_{i \in I} \sum_{l=1}^H \theta_l \prod_{u \in U} \prod_{r \in R} b_{iur}^{w_{iur}}. \tag{21}$$

Now, we estimate the $h_i(\Lambda)$ prior distribution of the Λ estimation parameter. The $h_i(\Lambda)$ prior probability distribution aims to model realistic considerations such as items' popularity information. To estimate $h_i(\Lambda)$, we employ the Dirichlet distribution as the conjugate prior for the $g_i(B|\Lambda)$ multinomial

Likelihood function. Because of the independency of parameters vectors θ and ω , the $h_i(\Lambda)$ prior distribution is deductively estimated as follows:

$$h_i(\Lambda) = h_i(\theta).h_i(\omega), \tag{22}$$

where $h_i(\theta)$ and $h_i(\omega)$ are partial prior distributions defined by Dirichlet functions.

The $h_i(\theta)$ distribution is computed as follows:

$$h_i(\theta) = \frac{1}{Be(\varphi)} \prod_{j=1}^K \prod_{l=1}^H \theta_l^{\varphi_j - 1}, \tag{23}$$

where $\varphi = (\varphi_1, \varphi_2, \dots, \varphi_K)$ are Beta function hyperparameters. For a non-informative prior, K and φ are set to 1.

The $h_i(\omega)$ distribution is obtained as follows:

$$h_i(\omega) = \frac{1}{Be(\varepsilon)} \prod_{j=1}^K \prod_{r \in R} b_{iur}^{\varepsilon_j - 1}, \tag{24}$$

where $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_K)$ are Beta function hyperparameters. For a non-informative prior, K and ε are set to 1.

After having computed the $g_i(B|\Lambda)$ Likelihood and the $h_i(\Lambda)$ prior distributions, we estimate the $g_i(\tilde{b}|B)$ Bayesian predictive distribution associated to an unknown observable \tilde{b} . For this purpose, we first infer the $g_i(\Lambda|B)$ posterior distribution of Λ estimation parameter given the B set of observations. Afterwards, from the $g_i(\Lambda|B)$ posterior distribution, the $g_i(\tilde{b}|B)$ Bayesian predictive distribution is inferred. Following the Bayes rule [49], the $g_i(\Lambda|B)$ posterior distribution is obtained as follows:

$$g_i(\Lambda|B) \propto h_i(\theta).h_i(\omega).g_i(B|\Lambda), \tag{25}$$

where $h_i(\theta)$ and $h_i(\omega)$ are prior distributions; and $g_i(B|\Lambda)$ is the Likelihood distribution.

The $g_i(\tilde{b}|B)$ Bayesian predictive distribution is computed as follows:

$$g_i(\tilde{b}|B) = \int_{\Lambda} g_i(\tilde{b}|\Lambda, B)g_i(\Lambda|B)d(\Lambda), \tag{26}$$

where $g_i(\Lambda|B)$ is the posterior distribution of the Λ estimation parameter given observations; and $g_i(\tilde{b}|\Lambda, B)$ is the joint sampling distribution for the \tilde{b} unknown observable given the B set of observations.

For a straight computation, the $g_i(\tilde{b}|B)$ distribution is approximated as follows:

$$g_i(\tilde{b}|B) = \sum_{l=1}^H g_i(\tilde{b}|\theta_l, B).g_i(\theta_l|B). \tag{27}$$

The $g_i(\tilde{b}|\theta_l, B)$ distribution component of Equation (27) is approximated as follows:

$$g_i(\tilde{b}|\theta_l, B) \propto \frac{\theta_l}{Be(\varepsilon)} \frac{\varepsilon + \sum_{j \in CLI_l} w_{jur}}{R\varepsilon + \sum_{r \in R} \sum_{j \in CLI_l} w_{jur}}. \tag{28}$$

The $g_i(\theta_l|B)$ distribution component of Equation (27) is computed as follows:

$$g_i(\theta_l|B) \propto \frac{1}{Be(\varphi)Be(\varepsilon)} \left(\frac{\varphi+1}{H\varphi+H} \right) \prod_{u \in U} \frac{\varepsilon + \sum_{j \in CLl} w_{jur}}{R\varepsilon + \sum_{r \in R} \sum_{j \in CLl} w_{jur}}. \quad (29)$$

Algorithm 2 details the Bayesian inference process regarding the item-based approach.

Algorithm 2 Item-Based BIHS Algorithm

Data:

- B : Tensor of probabilities of ratings on items
- θ : Vector of probabilities of items' cluster membership
- ω : Vector of ratings probabilities by items' cluster
- ε, φ : Beta function hyperparameters
- I : Set of items
- U : Set of users
- H : Number of items' clusters

Result: $g_i(\tilde{b}|B)$: The Bayesian predictive distribution for an unknown observable \tilde{b}

1 Begin

2 Compute the sampling distribution

$$g_i(B|\Lambda) = \prod_{i \in I} \sum_{l=1}^H \theta_l \prod_{u \in U} \prod_{r \in R} b_{iur}^{w_{iur}};$$

3 Compute the partial prior distribution $h_i(\theta)$ using (23);

4 Compute the partial prior distribution $h_i(\omega)$ using (24);

5 Compute the joint prior distribution $h_i(\Lambda) = h_i(\theta).h_i(\omega)$;

6 Compute the posterior distribution

$$g_i(\Lambda|B) = h_i(\Lambda).g_i(B|\Lambda);$$

7 for l **from** 1 **to** H **do**

8 $g_i(\tilde{b}|\theta_l, B) = \frac{\theta_l}{Be(\varepsilon)} \frac{\varepsilon + \sum_{j \in CLl} w_{jur}}{R\varepsilon + \sum_{r \in R} \sum_{j \in CLl} w_{jur}};$

9 $g_i(\theta_l|B) = \frac{1}{Be(\varphi)Be(\varepsilon)} \left(\frac{\varphi+1}{H\varphi+H} \right) \prod_{u \in U} \frac{\varepsilon + \sum_{j \in CLl} w_{jur}}{R\varepsilon + \sum_{r \in R} \sum_{j \in CLl} w_{jur}};$

10 Compute the Bayesian predictive distribution for an unknown observable \tilde{b}

$$g_i(\tilde{b}|B) += g_i(\tilde{b}|\theta_l, B).g_i(\theta_l|B);$$

11 end

12 Return $g_i(\tilde{b}|B)$;

The next subsection describes the approximation of our BIHS model which is a combination of the user-based approach and the item-based approach.

D. MARKOV CHAIN MONTE-CARLO SAMPLING

The above-computed Bayesian predictive distributions host a significant number of computationally greedy sums in high dimensions. In order to avoid computational overflows, we perform a Markov Chain Monte Carlo (MCMC) sampling of our model. MCMC sampling is a numerical integration

method that allows the approximation of an integral over continuous function by computing the function over a finite number of values [52]. The Metropolis algorithm is one of the most widely adopted MCMC sampling methods [48]. Like an adaptation of the Random Walker [53], it is based on conditioned moves towards the target approximate distribution. Iterations are performed in order to converge towards the target distribution. At each iteration, the move is validated by a x acceptance rate. For this purpose, from $g_u(\tilde{d}|\alpha_t, D)$, we draw a sample $\alpha^1, \alpha^2, \dots, \alpha^Q$; and from $g_i(\tilde{b}|\theta_l, B)$, we draw a sample $\theta^1, \theta^2, \dots, \theta^P$.

From the user-based approach, a move $\alpha_t \rightarrow \alpha_t^*$ is validated with a probability equal to $\min(x_u, 1)$, with the acceptance rate

$$x_u = \frac{g_u(\alpha_t^*|D)}{g_u(\alpha_t|D)} = \frac{\alpha_t^*(\mu + \sum_{j \in CLU_t} v_{jir}^*) R\mu + \sum_{r \in R} \sum_{j \in CLU_t} v_{jir}}{\alpha_t(\mu + \sum_{j \in CLU_t} v_{jir}) R\mu + \sum_{r \in R} \sum_{j \in CLU_t} v_{jir}^*}.$$

Recall that the variables v_{jir} and v_{jir}^* are respectively induced by α_t and α_t^* . The sample is used to approximate the $g_u(\tilde{d}|D)$ Bayesian predictive distribution as follows:

$$g_u(\tilde{d}|D) = \frac{1}{Q} \sum_{q=1}^Q g_u(\tilde{d}|\alpha^q, D). \quad (30)$$

Following the item-based approach, a move $\theta_l \rightarrow \theta_l^*$ is granted with a probability equal to $\min(x_i, 1)$, with the acceptance rate

$$x_i = \frac{g_i(\theta_l^*|B)}{g_i(\theta_l|B)} = \frac{\theta_l^*(\varepsilon + \sum_{j \in CLl} w_{jur}^*) R\varepsilon + \sum_{r \in R} \sum_{j \in CLl} w_{jur}}{\theta_l(\varepsilon + \sum_{j \in CLl} w_{jur}) R\varepsilon + \sum_{r \in R} \sum_{j \in CLl} w_{jur}^*}.$$

The random variables w_{jur} and w_{jur}^* are respectively induced by θ_l and θ_l^* . The sample is used to approximate the $g_i(\tilde{b}|B)$ Bayesian predictive distribution as follows:

$$g_i(\tilde{b}|B) = \frac{1}{P} \sum_{p=1}^P g_i(\tilde{b}|\theta^p, B). \quad (31)$$

The next subsection describes the rating prediction.

E. HYBRID RATING PREDICTION

The hybridization in our method consists of the merging of user-based and item-based approaches. The main advantage of using hybridization of those approaches is the reduction of the system sensitivity to the number of items that have been rated, and the number of users who have rated an item. This reduction of the system sensitivity contributes to the resilience of the system regarding the data sparseness problem.

The rating prediction process is based on both user and item-based approaches. The \tilde{p}_{ui} hybrid probability is calculated as follows:

$$\tilde{p}_{ui} = \left(g_i(\tilde{b}|B) \right)^{\frac{1}{|I|}} \cdot \left(g_u(\tilde{d}|D) \right)^{\frac{1}{|U|}}, \quad (32)$$

where $g_u(\tilde{d}|D)$ and $g_i(\tilde{b}|B)$ are the Bayesian predictive distributions respectively from the user-based approach and the item-based approach.

According to the MAP principle [35], the \tilde{r}_{ui} predicted rating is the rating that maximizes the \tilde{p}_{ui} predictive probability. It is obtained as follows:

$$\tilde{r}_{ui} = \arg \max_{r \in R} \tilde{p}_{ui}. \quad (33)$$

Algorithm 3 summarizes the hybrid Bayesian inference process.

Algorithm 3 BIHRS Algorithm

Data:

- D : Tensor of users' ratings probabilities
- B : Tensor of probabilities of ratings on items
- α : Vector of probabilities of users' cluster membership
- θ : Vector of probabilities of items' cluster membership
- γ : Vector of ratings probabilities by users' cluster
- ω : Vector of ratings probabilities by items' cluster
- $\mu, \varepsilon, \varphi, \lambda$: Beta function hyperparameters
- I : Set of items
- U : Set of users
- C : Number of users' clusters
- H : Number of items' clusters

Result: \tilde{r}_{ui} the predicted rating

1 Begin

- 2 Compute the $g_u(\tilde{d}|D)$ Bayesian predictive distribution for an unknown observable \tilde{d} using Algorithm 1;
- 3 Compute the $g_i(\tilde{b}|B)$ Bayesian predictive distribution for an unknown observable \tilde{b} using Algorithm 2;
- 4 MCMC model approximation;
- 5 # Hybrid rating prediction
- 6 Compute the predicted rating

$$\tilde{r}_{ui} = \arg \max_{r \in R} \left[\left(g_i(\tilde{b}|B) \right)^{\frac{1}{|I|}} \cdot \left(g_u(\tilde{d}|D) \right)^{\frac{1}{|U|}} \right];$$

- 7 Return \tilde{r}_{ui} ;
-

Items are ordered from the most liked by the target user u to the least liked. The *Top-N* set of items with the highest ratings is the list of items that will be recommended to the target user.

F. DRIVING EXAMPLE

In this section, we run an example to show how our proposal works. Tab. 1 on which we apply our method, presents the

TABLE 1. Data Matrix.

Matrix	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	5	?	3	1	4	2	1	2	1	3
u_2	2	4	5	3	1	?	2	5	4	?
u_3	1	3	?	4	2	5	4	?	2	1
u_4	3	2	1	5	?	4	5	3	3	5
u_5	4	5	2	2	3	1	2	1	3	4
u_6	2	1	4	?	5	3	3	4	5	2

data matrix that hosts ratings assigned by six users on ten items.

We run our method under the following settings:

- The number of users' cluster is fixed to one ($t = 1$). This means that there is only one users' cluster. The cluster membership probability is the same for all users of the cluster and the cluster membership probability is evaluated as $\alpha = \alpha_t = 1$.
- The number of items' cluster is set to one ($l = 1$). It means that there is only one items' cluster. The cluster membership probability is the same for all items belonging to the cluster and is estimated as $\theta = \theta_l = 1$.
- From the user-based approach, the hyperparameter λ in Equation (7) is set to 1 for a non-informative prior probability distribution. Deductively, we compute the users' preference probability in Equation (7) as $h_u(\alpha) = 1$.
- From the item-based approach, the hyperparameter φ of the $h_i(\theta)$ preference probabilities on items in Equation (23) is set to 1 for a non-informative prior probability distribution. Deductively, we find $h_i(\theta) = 1$.
- We consider the set of possible values of ratings $R = \{1; 2; 3; 4; 5\}$.

TABLE 2. Likelihood probability according to the user-based approach.

	Possible values of Ratings					Likelihood probability
	1	2	3	4	5	
i_1	0.1667	0.3334	0.1667	0.1667	0.1667	2.572e-4
i_2	0.2	0.2	0.2	0.2	0.2	3.2e-4
i_3	0.2	0.2	0.2	0.2	0.2	3.2e-4
i_4	0.2	0.2	0.2	0.2	0.2	3.2e-4
i_5	0.2	0.2	0.2	0.2	0.2	3.2e-4
i_6	0.2	0.2	0.2	0.2	0.2	3.2e-4
i_7	0.1667	0.3334	0.1667	0.1667	0.1667	2.572e-4
i_8	0.2	0.2	0.2	0.2	0.2	3.2e-4
i_9	0.1667	0.1667	0.3334	0.1667	0.1667	2.572e-4
i_{10}	0.2	0.2	0.2	0.2	0.2	3.2e-4

TABLE 3. Likelihood probability according to the item-based approach.

Rating	u_1	u_2	u_3	u_4	u_5	u_6
1	0.3334	0.125	0.250	0.1112	0.2	0.1112
2	0.2223	0.250	0.250	0.2223	0.3	0.2223
3	0.2223	0.125	0.125	0.3334	0.2	0.2223
4	0.1112	0.250	0.250	0.1112	0.2	0.2223
5	0.1112	0.250	0.125	0.2223	0.1	0.2223
Likelihood	2.032e-4	2.441e-4	2.441e-4	2.032e-4	2.4e-4	2.709e-4

We aim to predict u_5 's interest concerning item i_3 . Knowing the observed rating assigned by user u_5 on i_3 , the purpose is to estimate using our method the u_5 's preference degree on item i_3 . Tab. 2 and Tab. 3 show Likelihood probabilities

according to user-based and item-based approaches respectively. In Tab. 2, using Equation (5), the Likelihood probability on i_3 is computed as follows:

$$\prod_{r=1}^5 \frac{\text{Number of users having assigned } r \text{ on item } i_3}{\text{Number of users having rated item } i_3} = \frac{1}{5} * \frac{1}{5} * \frac{1}{5} * \frac{1}{5} * \frac{1}{5} = 3.2e - 4 \quad (34)$$

Following the item-based approach in Tab. 3, using Equation (21), the Likelihood probability of the u_5 's interest is computed as follows:

$$\prod_{r=1}^5 \frac{\text{Number of items rated } r \text{ by user } u_5}{\text{Number of items rated by } u_5} = \frac{2}{10} * \frac{3}{10} * \frac{2}{10} * \frac{2}{10} * \frac{1}{10} = 2.4e - 4 \quad (35)$$

Tab. 4 shows the Bayesian predictive probability with which user u_5 rates item i_3 following the user-based, item-based and the hybrid approaches. For instance, using Equation (18), the Bayesian predictive probability with which user u_5 rates item i_3 with the rating $r = 1$ following the user-based approach is estimated as follows:

$$\prod_{i=i_1(i \neq i_3)}^{i_{10}} \frac{\text{Number of users having assigned } r = 1 \text{ on item } i}{\text{Number of users having rated item } i} = \frac{1}{6} * \frac{1}{5} * \frac{1}{5} * \frac{1}{5} * \frac{1}{5} * \frac{1}{6} * \frac{1}{5} * \frac{1}{6} * \frac{1}{5} = 2.963e - 7 \quad (36)$$

Using Equation (29), the Bayesian predictive probability with which user u_5 rates item i_3 with the rating $r = 1$ following the item-based approach is estimated as follows:

$$\prod_{u=u_1(u \neq u_5)}^{u_6} \frac{\text{Number of items rated } r = 1 \text{ by user } u}{\text{Number of items rated by user } u} = \frac{3}{9} * \frac{1}{8} * \frac{2}{8} * \frac{1}{9} * \frac{1}{9} = 1.286e - 4 \quad (37)$$

Using Equation (32), the Bayesian predictive probability with which user u_5 rates item i_3 with the rating $r = 1$ following the hybrid approach is estimated as follows:

$$(2.963e - 7)^{\frac{1}{10}} (1.286e - 4)^{\frac{1}{5}} = 0.03706 \quad (38)$$

Regarding Tab. 4, it can be observed that the rating value $r = 2$ by user u_5 on item i_3 shows the highest probability. In other words, the rating value $r = 2$ is the predicted rating following our method. This estimation perfectly matches with the observation in Tab. 1 concerning u_5 's interest about i_3 as he really assigned rating $r = 2$ to item i_3 in Tab. 1.

TABLE 4. Prediction of user u_5 on item i_3 .

	Possible values of Ratings				
	1	2	3	4	5
User-based approach	2.963e-7	1.185e-6	5.925e-7	2.962e-7	2.962e-7
Item-based approach	1.286e-4	6.858e-4	2.572e-4	1.715e-4	1.715e-4
Hybrid approach	0.03706	0.0595	0.0456	0.03926	0.03926

G. BIHRS RECOMMENDATION UNDERSTANDING

Our BIHRS model is a hybridization that combines the user-based approach hereafter called User-based BIHRS and the item-based approach hereafter called Item-based BIHRS.

Unlike baseline model-based CFs, our approach is realistic and easier to understand thanks to the prior splitting into clusters of users and items. Indeed, following a K-Nearest Neighbors (KNN) reasoning [7], users as much as items necessarily belong to a cluster. In addition, our model integrates a priori information such as the items' popularity and the most frequent users' needs. Indeed, the prior probability distributions $h_u(\Omega)$ and $h_i(\Lambda)$ model a priori intuitions that can improve the quality of the estimate. Faced with a multitude of items, users only rate a small number of them. The prediction is therefore performed to estimate the missing features.

Following the User-based BIHRS approach, the recommendation is understandable as being based on the probability with which a user u would rate an item i given the D set of observations. In other words, a user u is most likely to like items that have a positive and high popularity with other users of the same clusters as u . If in addition to the D set of observations, a priori information referring to the item reputation are known, then our model integrates these information in order to refine the estimate.

Regarding the Item-based BIHRS approach, the recommendation is explainable as being based on the probability for an item i to be liked by a user u given the B set of observations. In other words, the item i is most likely to be liked by users who previously liked similar items belonging to the same cluster. If in addition to the B set of observations, a priori information related to the reputation of the most recurrent users' needs are known, then, our model includes these data in order to improve the prediction quality.

The hybrid BIHRS approach is therefore explainable as being based on the probability with which a user u would like an item i given the evidence relative to users belonging to the same cluster and having previously liked the item i , and liked items similar belonging to the same cluster as i .

The next section presents the experimentations and results.

IV. EXPERIMENTATIONS AND RESULTS DISCUSSION

In this section, we assess our method performances comparatively to other recommendation methods. Experiments are

TABLE 5. Datasets Specifications.

Dataset	Number of Users	Number of Items	Number of Ratings
MovieLens1M	6.040	3.900	1.000.209
MovieLens10M	280.000	58.000	27.000.000
Flixster	147.612	48.794	8.196.077
Ciao	7.375	99.746	278.483
Epinions	40.163	139.738	664.824

conducted on open real-world datasets from MovieLens,¹ Ciao², Epinions² and Flixster.² The features of those datasets are specified in Tab. 5.

The next subsection details the experiment process.

A. EXPERIMENTS SETUP

Experiments are conducted on a computer hosting a processor of type Intel Core i7 (2.4 GHz) with 16 GB RAM, running Windows 10 Operating System. The free-available CF4J framework [54] has been used to implement our algorithm. We have used Netbeans version 8.2 as Java development environment.

The used datasets host invocations of 3.900 movies by 6.040 users generating 1.000.209 ratings regarding the MovieLens1M dataset and 58.000 movies invocated by 280.000 users generating 27.000.000 ratings following the MovieLens10M dataset. In Flixster dataset, 147.612 users rate 48.794 items and generate 8.196.077 ratings; while 40.163 users interact on 139.738 items via 664.824 ratings in Epinions dataset. Ciao dataset hosts 7.375 users which rate 99.746 items and generate 278.483 ratings. Each rating is a value ranging from 1 (referring to a poor user satisfaction) to 5 (meaning a high satisfying user experience).

For each dataset, a rating matrix is built. The rating matrix is split into test data and training data. We define the data sparsity as the density of the data matrix. It describes the proportion of data used for the recommendation process. The matrix is progressively dug to sketch the data sparsity. For this purpose, the matrix sparsity also known as the matrix porosity intensity ranges from 20% to 80% in steps of 20%. For instance, a matrix sparsity set to 20% means that datasets are divided into 20% of test data and 80% of training data.

The next subsection shows the evaluation metrics used to assess our method performances.

B. EVALUATION METRICS

The BIHRS's prediction accuracy is assessed using key indicators namely the Mean Absolute Error (MAE) and the Root Square Mean Error (RSME). The Normalized Discounted Cumulative Gain (NDCG) is additionally used to assess the ranking accuracy. Low values of MAE and RSME express a high prediction accuracy. MAE and RSME values are

computed as follows:

$$MAE = \frac{\sum_{s \in S_{rec}} |r_{ui} - \tilde{r}_{ui}|}{N}, \quad (39)$$

$$RSME = \sqrt{\frac{\sum_{s \in S_{rec}} (r_{ui} - \tilde{r}_{ui})^2}{N}}, \quad (40)$$

where r_{ui} is the original rating and \tilde{r}_{ui} the approximated value; N is the number of recommendations and S_{rec} is the set of recommendations.

NDCG is widely used to assess the ranking accuracy and is computed as follows:

$$NDCG_N = \frac{DCG_N}{IDCG_N}, \quad (41)$$

where $IDCG_N$ and DCG_N respectively refer to the Ideal Discounted Cumulative Gain and the Discounted Cumulative Gain of Top- N recommended items. DCG_N is computed as follows:

$$DCG_N = \sum_{j=1}^N \frac{2^{rel_j} - 1}{\log_2(j + 1)}, \quad (42)$$

where rel_j is the rating associated to the item ranked at position j . A high $NDCG_N$ means a high-accurate ranking.

The recommendation quality is assessed using the precision and the recall indicators. The high precision and recall values express a high recommendation quality. The recall indicator measures the proportion of items correctly recommended comparatively to the number of items expected to be recommended. Meanwhile, the precision indicator measures the proportion of items correctly recommended comparatively to the number of items recommended. The precision and recall measures are computed as follows:

$$Precision = \frac{|S \cap E|}{S}, \quad (43)$$

$$Recall = \frac{|S \cap E|}{E}, \quad (44)$$

where S is the set of recommended items and E the set of items expected to be recommended.

C. RESULTS AND ANALYSIS

The datasets that we use do not provide a priori information such as the movies' popularity or the consumption habits of a users' majority. For this reason, for non-informative prior probability distributions, their hyperparameters have been set to 1.

In the following points, BIHRS's performances are evaluated comparatively to other recommendation methods.

1) PERFORMANCES ANALYSIS

Referring to the distribution of users and items in clusters, our model is based on a reasoning inspired by the KNN algorithm. Furthermore, our method is a precise model-based CF performing an understandable prediction. For this reason, its

¹<https://grouplens.org/datasets/movielens/>

²<https://www.librec.net/datasets.html>

TABLE 6. Data Sparsity Impact on BIHRS Prediction Accuracy.

Matrix Sparsity (%)		20			40			60			80		
Datasets	Algorithms	MAE	RMSE	NDCG	MAE	RMSE	NDCG	MAE	RMSE	NDCG	MAE	RMSE	NDCG
Movielens	UPCC	0.856	1.066	0.611	0.868	1.089	0.587	0.887	1.117	0.564	0.970	1.170	0.463
	IPCC	0.742	0.909	0.683	0.953	1.197	0.602	1.023	1.274	0.499	1.253	1.439	0.270
	Spearman-based CF	0.771	0.961	0.724	0.785	0.986	0.683	0.809	1.022	0.663	0.909	1.022	0.563
	BIHRS	0.674	0.740	0.926	0.778	0.958	0.698	0.791	0.978	0.688	0.840	0.991	0.617
	NBCF	0.736	0.861	0.887	0.783	0.965	0.683	0.795	0.995	0.648	0.855	0.997	0.597
	PMF	0.740	0.875	0.874	0.779	0.960	0.663	0.797	1.006	0.643	0.866	1.008	0.570
	BNMF	0.749	0.897	0.855	0.782	0.962	0.653	0.802	1.012	0.655	0.879	1.015	0.567
Flixster	UPCC	1.768	1.899	0.653	1.787	1.969	0.546	1.794	1.976	0.357	1.795	2.085	0.204
	IPCC	1.376	1.893	0.643	1.578	1.909	0.536	1.774	1.926	0.352	1.783	1.989	0.234
	Spearman-based CF	1.759	1.881	0.654	1.765	1.903	0.549	1.769	1.920	0.354	1.779	1.982	0.231
	BIHRS	0.874	0.969	0.889	0.883	0.992	0.757	0.898	0.995	0.651	0.931	1.099	0.530
	NBCF	0.947	1.086	0.847	0.966	1.118	0.626	1.011	1.169	0.675	1.002	1.186	0.565
	PMF	0.984	1.122	0.868	0.991	1.293	0.658	0.998	1.341	0.587	1.088	1.372	0.585
	BNMF	1.090	1.335	0.862	1.150	1.590	0.677	1.230	1.685	0.699	1.550	1.728	0.513
Ciao	UPCC	1.216	1.430	0.983	1.248	1.547	0.702	1.295	1.687	0.479	1.455	1.714	0.476
	IPCC	1.207	1.488	0.979	1.252	1.514	0.717	1.280	1.628	0.679	1.452	1.794	0.425
	Spearman-based CF	1.202	1.471	0.970	1.251	1.617	0.899	1.291	1.669	0.785	1.446	1.780	0.545
	BIHRS	0.823	0.922	0.989	0.842	0.977	0.817	0.875	0.981	0.747	0.882	1.031	0.562
	NBCF	0.900	1.080	0.955	0.969	1.155	0.779	0.974	1.263	0.648	0.927	1.350	0.532
	PMF	0.915	1.198	0.924	0.976	1.293	0.776	1.080	1.349	0.540	1.165	1.459	0.512
	BNMF	1.034	1.399	0.972	1.131	1.491	0.739	1.211	1.507	0.658	1.374	1.605	0.527
Epinions	UPCC	1.674	1.890	0.788	1.707	1.924	0.512	1.827	2.267	0.506	1.900	2.317	0.415
	IPCC	1.592	1.730	0.793	1.609	1.754	0.591	1.965	2.074	0.569	1.842	2.142	0.503
	Spearman-based CF	1.521	1.772	0.761	1.788	1.814	0.750	1.829	2.116	0.501	1.856	2.076	0.414
	BIHRS	0.993	1.189	0.836	1.113	1.250	0.701	1.129	1.320	0.630	1.275	1.414	0.575
	NBCF	1.121	1.473	0.811	1.188	1.514	0.750	1.301	1.529	0.516	1.331	1.576	0.504
	PMF	1.126	1.498	0.806	1.193	1.538	0.759	1.364	1.623	0.584	1.393	1.695	0.518
	BNMF	1.376	1.583	0.813	1.431	1.625	0.709	1.718	1.928	0.528	1.890	1.975	0.591

performances are evaluated comparatively to neighborhood-based methods also known as memory-based approaches, and to model-based methods. Those methods are hereafter presented:

- IPCC is a memory-based method that lays on the inter-item similarities computation [12] using Pearson Correlation Coefficient (PCC). In this method, a neighborhood defined as a collection of items having been rated by the same users is built. Thereafter, the predicted rating is computed as a weighted mean of ratings over items belonging to the neighborhood.
- UPCC is a memory-based method that lays on the inter-user similarities computation [8] using PCC. In this approach, a neighborhood defined as a set of users having rated the same items is built. Thereafter, the predicted rating is computed as a weighted mean of ratings assigned by users belonging to the neighborhood.
- Spearman-based CF is a memory-based method that performs inter-user and inter-item similarities computations [15] using Spearman Rank Correlation Coefficient (SRCC). Afterwards, the prediction is made based on user and item approaches.
- Bayesian Non-Negative Matrix Factorization (BNMF) based method is a model-based method. It is a probabilistic recommendation approach based on a non-negative variant of matrix factorization [20]. This approach decomposes the rating matrix into non-negative latent features matrices. Users and items are divided into groups in order to increase the efficiency

of the matrix factorization. The BNMF method is parametrized as follows:

- The degree of evidence needed to the users’ ratings inference is set to 4.
- The number of iterations to the algorithm convergence is set to 150.
- The number of latent factors is set to 12.
- Probabilistic Matrix Factorization (PMF) based method is a model-based recommendation method built on the baseline matrix factorization following a probabilistic approach [29]. This method decomposes the high-dimensional rating matrix into low-dimensional latent features matrices. Thereafter, the Gradient optimization method is performed to determine the optimal low-dimension matrices. The PMF method is parametrized as follows:
 - The number of iterations for the convergence of the algorithm is set to 150.
 - The step size of an iteration is set to 0.05.
 - The number of latent features is set to 10.
- Naive Bayesian Collaborative Filtering (NBCF) method is a recommendation approach based on the Naive Bayes classifier in which uniform probability distributions are used and therefore simplify the modeling of interactions between users and items [35].

2) ASSESSMENT OF THE PREDICTION ACCURACY

Following Tab. 6, the prediction quality of our method is compared to that of the existing model and memory-based

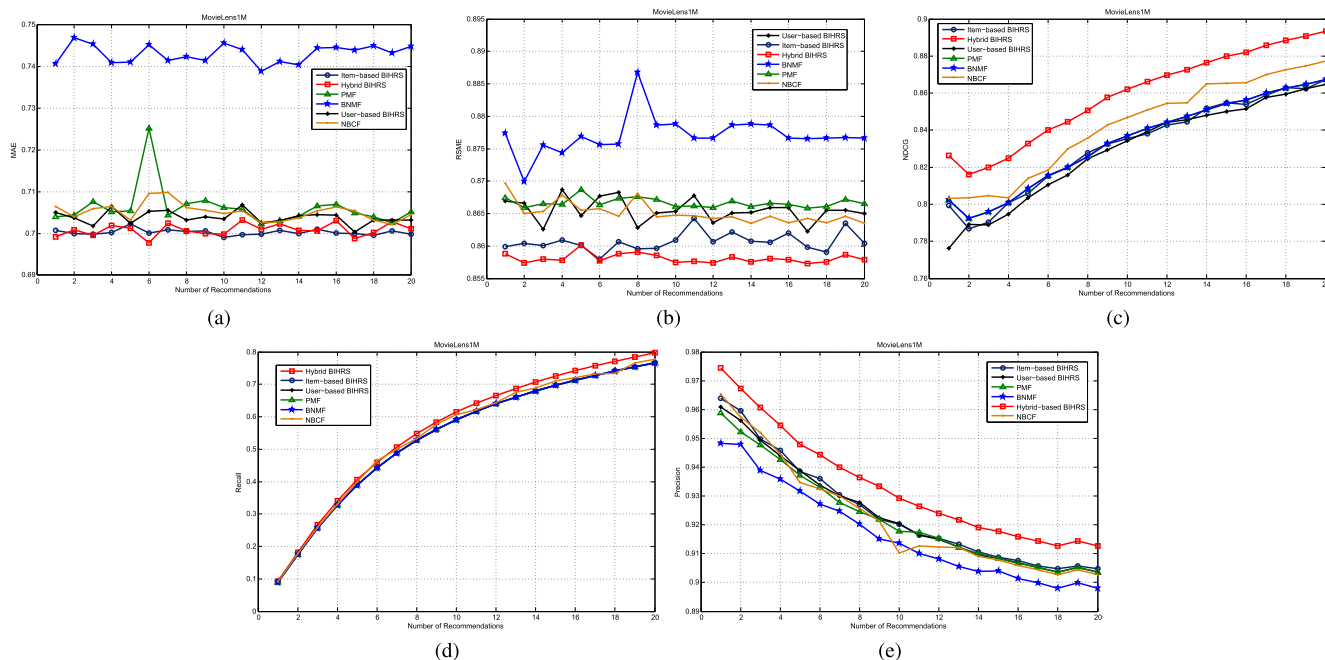


FIGURE 3. BIHRS Performances on MovieLens1M Dataset.

recommendation methods. It can be observed that the prediction accuracy of our method outperforms that of competing model and memory-based approaches. Indeed, MAE and RSME performances of our BIHRS model are lower than those of IPCC, UPCC, Spearman-based CF, NBCF, PMF and BNMF. This effectiveness of our method is due to the fact that our approach is built on a model that learns from users’ and items’ observations to thereafter infer users’ behaviors based on realistic considerations such as users’ intuitions and items’ features. Furthermore, our approach shows a higher NDCG trend compared to that of other recommendation methods. It means that regarding the ranking accuracy, our proposal is better than other methods. In addition, it can be observed that the matrix sparsity affects globally the performances of each method indicated in Tab. 6 due to the fact that the increase of matrix sparsity rate induces a decrease in training data that feed each recommendation approach. However, despite that negative impact, our approach outperforms other methods and thus highlights its data sparsity resiliency capacity.

Compared to model-based recommendation methods, Fig. 3a depicts the MAE performance of our proposal from the usage of the MovieLens1M dataset. It can be observed that MAE trends of our method and its two variants namely User-based BIHRS and Item-based BIHRS are the lowest, meaning a high prediction accuracy compared to that of other model-based methods. This prediction accuracy improvement is explained by the fact that by modeling plausible intuitions or a priori information, our method includes more realistic aspects in the prediction model than other model-based methods.

Fig. 4a depicts the MAE performance evaluation from the usage of the MovieLens10M dataset. It can be observed that the MAE trend of our method is globally improved from 0.7 to 0.692 on average. This is due to the fact that the MovieLens10M dataset hosts more ratings than the MovieLens1M dataset. Hence, by using the MovieLens10M dataset, many more observations are involved in the inference process to refine the prediction. Once more, our method shows a MAE performance better than that of other model-based approaches. In Fig. 5a, 6a and 7a, our method performances are shown using Flixster, Ciao, and Epinions datasets. It can be observed that MAE performances of our method are lower and therefore better than those of other methods.

From the usage of the MovieLens1M dataset, Fig. 3b depicts the RSME performance of our approach compared to model-based recommendation methods. It can be observed that our method and its two variants show a lower RSME trend meaning that our approach prediction accuracy is better than that of other model-based methods.

Fig. 4b shows the RSME performance from the usage of the MovieLens10M dataset. It can be noted that thanks to more data in the dataset, the RSME performance of our method is roughly improved from 0.857 to 0.845 on average. Hence, by its RSME trend, our method outperforms other model-based approaches. In Fig. 5b, 6b and 7b, our method performances are observed using Flixster, Ciao, and Epinions datasets. It can be observed that RSME performances of our proposal are lower and therefore better than those of other methods.

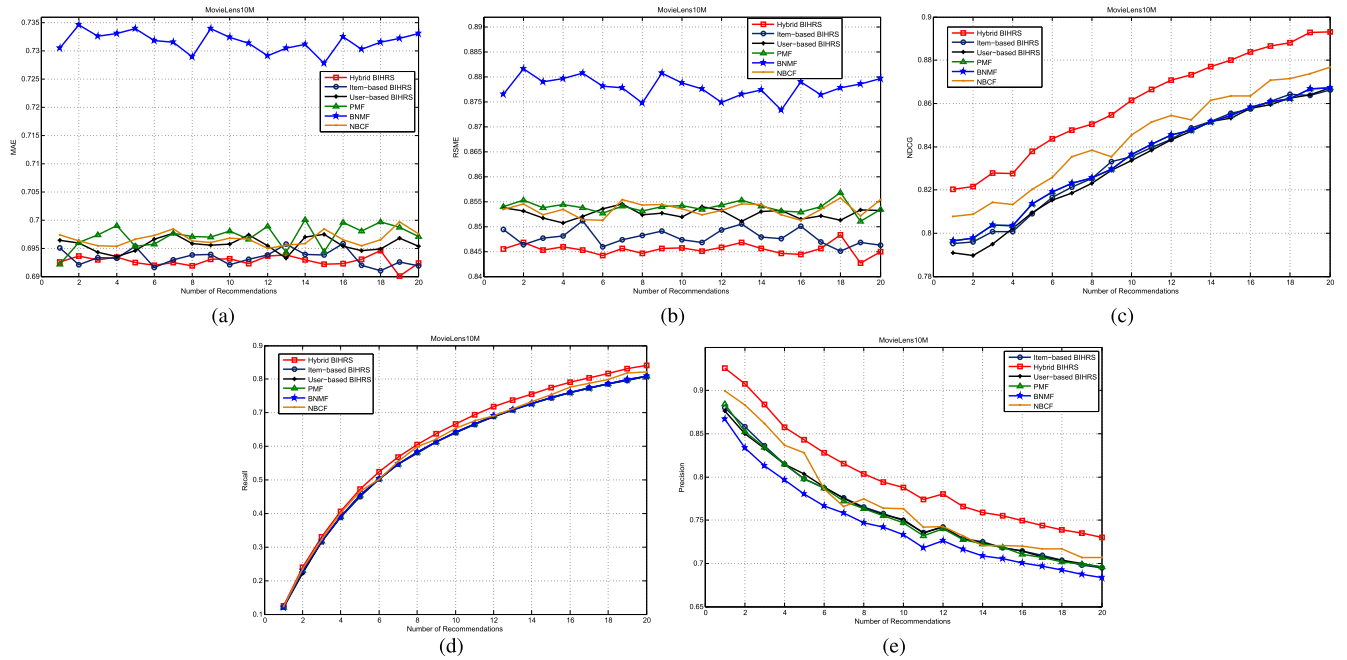


FIGURE 4. BIHRS Performances on MovieLens10M Dataset.

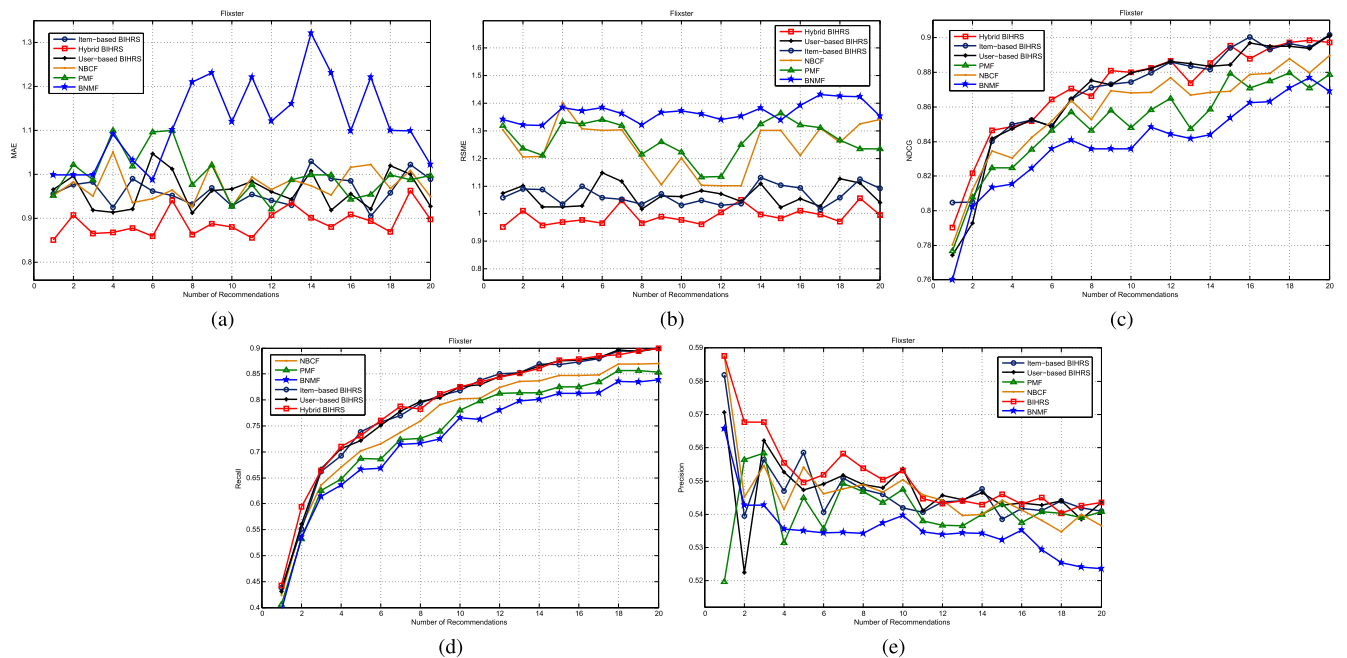


FIGURE 5. BIHRS Performances on Flixster Dataset.

From the usage of the MovieLens1M dataset, Fig. 3c shows the NDCG performance of our proposal compared to model-based recommendation methods. It can be observed that our method shows a higher NDCG trend meaning a ranking accuracy higher than that of other methods.

Exploiting the MovieLens10M dataset, Fig. 4c shows the NDCG improvement of our proposal and its variants

compared to other methods. Indeed, it can be highlighted a NDCG increase of 2.5% on average compared to other model-based methods. In Fig. 5c, 6c and 7c, our method performances are shown using Flixster, Ciao, and Epinions datasets. It can be observed that NDCG performances of our approach are higher and therefore better than those of other methods.

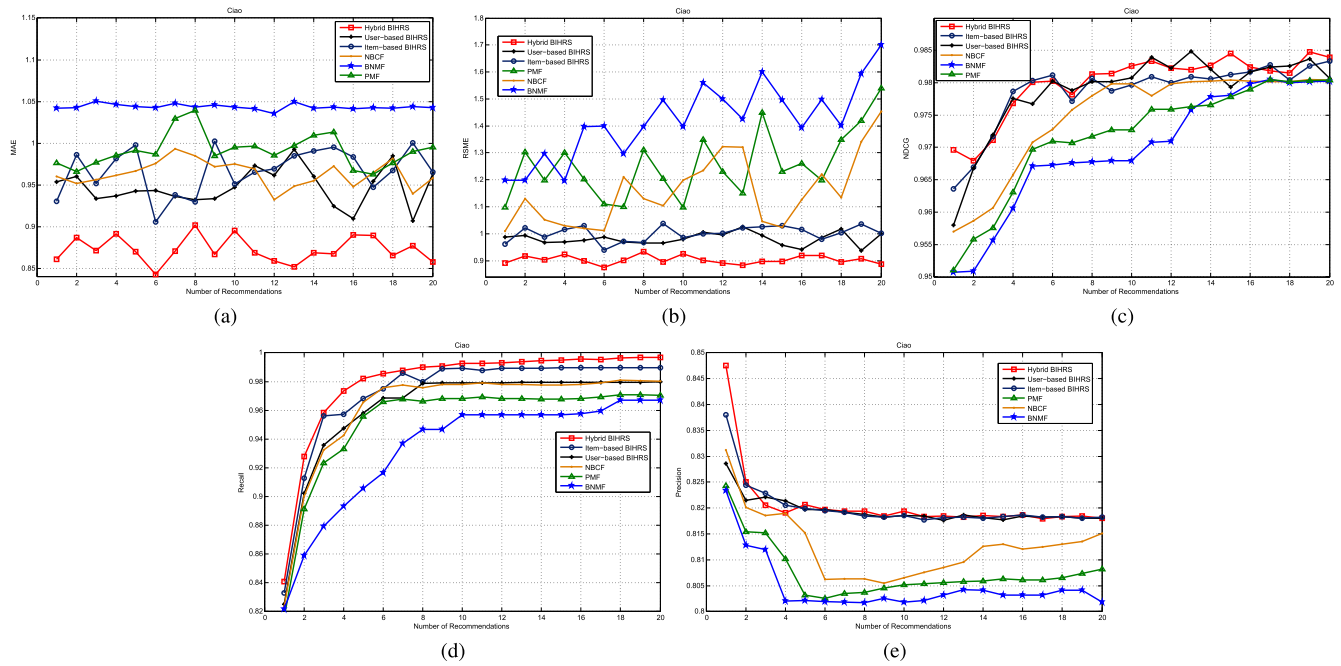


FIGURE 6. BIHRS Performances on Ciao Dataset.

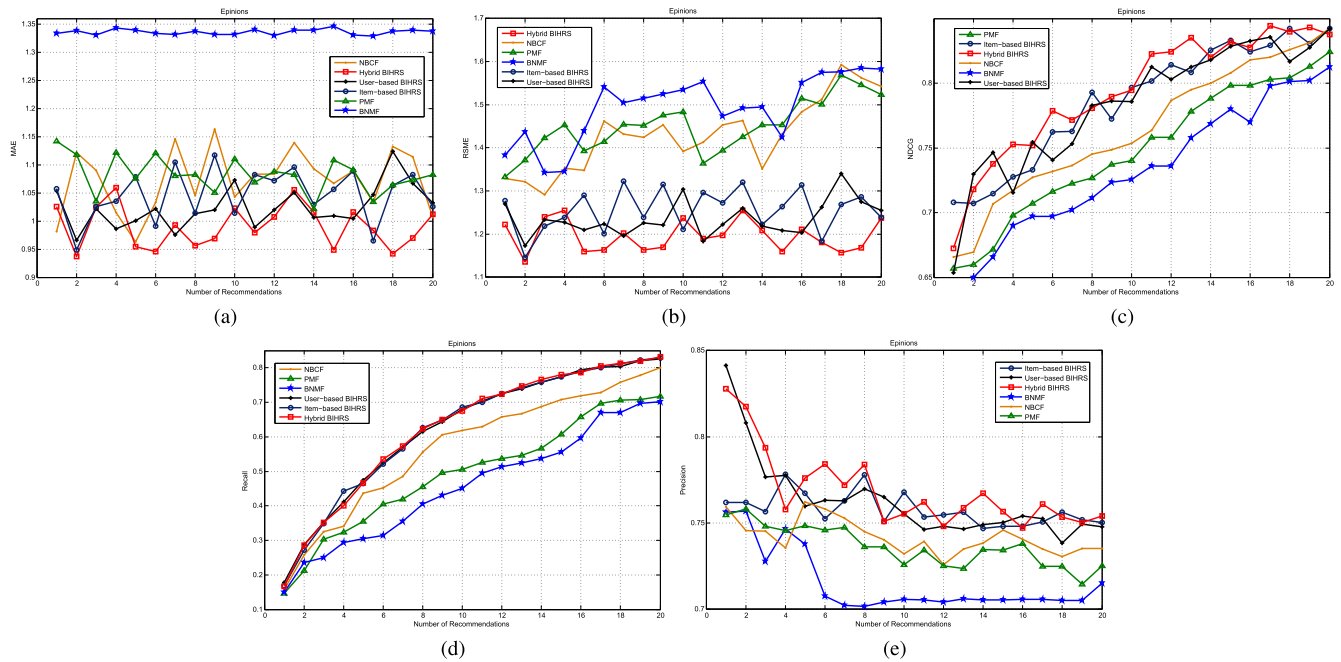


FIGURE 7. BIHRS Performances on Epinions Dataset.

3) EVALUATION OF THE RECOMMENDATION QUALITY

According to Tab. 7, the recommendation quality of our method is evaluated comparatively to that of the competing memory-based recommendation methods. It can be observed that, for each method mentioned in Tab. 7, the precision is globally affected by the increase in the number of recommendations. It is explained by the fact that the increasing

number of recommendations induces the addition of low prediction accuracy items in the recommendations list and consequently affects the recommendation precision. The recall trend of all methods increases with the number of recommendations because a significant recommendation number increases the possibility of recommending expected items. Despite this fact, the recommendation quality of our method

TABLE 7. BIHRS Recommendation Quality Compared to Baseline Memory-based CFs.

Number of Recommendations		5		10		15		20	
Datasets	Algorithms	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Movielens	UPCC	0.348	0.401	0.338	0.585	0.330	0.676	0.324	0.732
	IPCC	0.365	0.091	0.347	0.451	0.338	0.609	0.327	0.687
	Spearman-based CF	0.417	0.493	0.364	0.687	0.335	0.789	0.315	0.847
	BIHRS	0.937	0.505	0.917	0.694	0.908	0.825	0.903	0.897
Flixster	UPCC	0.512	0.346	0.452	0.523	0.306	0.612	0.264	0.752
	IPCC	0.536	0.264	0.482	0.354	0.343	0.654	0.315	0.692
	Spearman-based CF	0.504	0.453	0.454	0.642	0.421	0.726	0.397	0.783
	BIHRS	0.549	0.731	0.553	0.825	0.546	0.876	0.543	0.899
Ciao	UPCC	0.534	0.268	0.425	0.447	0.336	0.568	0.321	0.781
	IPCC	0.401	0.275	0.395	0.468	0.384	0.634	0.338	0.741
	Spearman-based CF	0.484	0.510	0.332	0.641	0.315	0.754	0.302	0.865
	BIHRS	0.821	0.982	0.819	0.992	0.818	0.994	0.818	0.997
Epinions	UPCC	0.544	0.482	0.351	0.589	0.343	0.634	0.302	0.745
	IPCC	0.468	0.151	0.364	0.435	0.342	0.534	0.312	0.654
	Spearman-based CF	0.565	0.415	0.541	0.645	0.452	0.757	0.384	0.844
	BIHRS	0.776	0.466	0.755	0.674	0.756	0.779	0.754	0.832

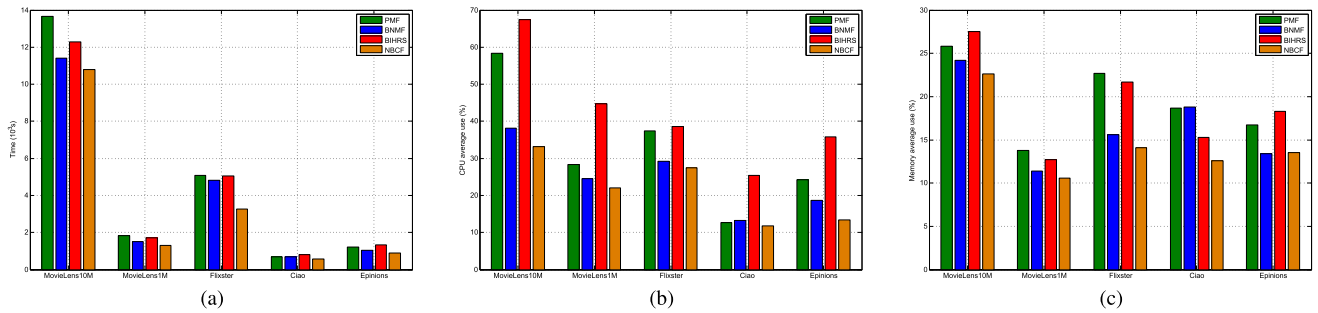


FIGURE 8. Execution time, CPU and Memory use.

remains better than that of memory-based approaches (IPCC, UPCC, and Spearman-based CF) since the precision of our BIHRS method is the highest. Indeed, referring to Tab. 7, the recommendation quality of our method is better than that of memory-based approaches since the precision and recall trends of our BIHRS method are higher than those of IPCC, UPCC and Spearman-based CF.

Now, we compare our method to other model-based methods. Fig. 3e, 5e, 6e and 7e depict the recommendation precision of our proposal using MovieLens1M, Flixster, Ciao and Epinions datasets. The precision trend is affected by the increase in the number of recommendations due to low-accurately predicted items added to the set of recommendations. However, it can be observed that the precision shown by our method is better than that of other model-based methods. This precision improvement is justified by the fact that our model integrates more features in the prediction process than other methods.

From the usage of the MovieLens10M dataset, Fig. 4e highlights the recommendation precision of our proposal. It can be observed that our method is scalable because the BIHRS’s precision is higher than that of other model-based methods, even in the case of large datasets.

Fig. 3d, 5d, 6d and 7d highlight the recall trends of our proposal compared to other model-based recommendation

methods using MovieLens1M, Flixster, Ciao and Epinions datasets. It can be observed that our recall performance method is improved compared to other methods.

Exploiting the MovieLens10M dataset, Fig. 4d shows the recall performance of our proposal which is slightly enhanced compared to other model-based recommendation methods. In addition, compared to BIHRS’s recall trend obtained by using the MovieLens1M dataset, it can be observed that the scalability of our method is highlighted because the BIHRS’s recall trend obtained by using the MovieLens10M dataset is increased of 3% on average.

4) COMPUTATIONAL RESOURCES CONSUMPTION

Fig. 8 shows the computational resources consumption of each model-based methods. In Fig. 8a, it can be observed that our method shows a competitive execution time slightly higher than that of NBCF and BNMf methods but lower than that of PMF for the large MovieLens10M dataset. The relatively low execution time of the BIHRS method highlights the scalability of our proposal for large datasets. For Ciao and Epinions datasets, our method’s time execution is slightly higher but it is offset by significant recommendation performances of BIHRS compared to other model-based methods. Fig. 8b and Fig. 8c show that our proposal consumes

more Central Processing Unit (CPU) and memory resources than other methods. It is due to the complexity of computed non-uniform probability distributions. However, it has been proved that our method shows an increased prediction accuracy (see Fig. 3, 4, 6, 7, 5) and an enhanced recommendation quality (see Tab. 3 and Tab. 4) compared to other methods. For this reason, our method appears as a valuable trade-off between a high-accurate recommendation and an affordable computational cost.

5) THREATS TO VALIDITY

In our proposal, the recommendation is performed on datasets used to model users' tastes in the training phase. Experiments have been conducted on datasets with significant matrix density and in which the number of items jointly rated by users is significant. Consequently, the evaluation of users' tastes similarity is accurate since users express their interest in the same items. In the case of datasets in which users do not jointly rate items, the assessment of users' preferences proximity could be corrupted and therefore affect the prediction accuracy due to clusters hosting dissimilar users. In other words, the recommendation precision could be threatened in the case of datasets in which users do not jointly rate a significant number of items. In addition, users' and items' cluster sizes have to be carefully chosen since small cluster sizes could affect the recommendation accuracy as probability distributions are computed on clusters, while big cluster sizes could noise the prediction process with additional irrelevant information. Knowing that the number of clusters is set at the initialization of the clustering process, it should be chosen with caution since a small number of clusters could induce big cluster sizes in which there will be noisy data; while a high number of clusters could induce small cluster sizes in which data will be sparse. However, our proposal remains general since experimentations have been conducted on popular and widely adopted datasets in the recommendation field.

V. CONCLUSION AND PERSPECTIVES

In this paper, we have addressed the item targeting problem by proposing a hybrid model-based recommendation approach that is built on the Bayesian inference in order to predict users' interests. The proposed method excels through its easy-explainability like that of memory-based CFs, and its high-precision like model-based CFs. The Bayesian prediction model developed in our proposal employs non-uniform probability distributions to model intuitions, and a priori information related to items' features and users' behaviors. The proposed method merges a user-based approach and an item-based approach to alleviate the data sparsity, and therefore enhance the prediction accuracy by exploiting all various aspects raised by user-item interactions. Afterwards, the MAP principle helps to predict user's interest in a given item by computing the rating that maximizes the Bayesian predictive distribution on the unknown observable. Experiments have been conducted on free-available datasets and show that the proposed method outperforms others

(UPCC, IPCC, Spearman-based CF) regarding the MAE and the RSME performances. Besides, the results obtained highlight the improved prediction accuracy of our proposal, the ranking accuracy of the proposed method through its valuable NDCG performances. Moreover, the effectiveness of our approach expressed by its increased prediction accuracy and its improved recommendation quality has been shown comparatively to other model-based recommendation methods (PMF, NBCF, BNMF).

Given the rapid growth of Big Data, an extension of our approach could be developed by the integration of other sources of information. Indeed, the high intensity of users' activity on social networks offers increased possibilities and rich sources of information to embed in the recommendation process. For this purpose, to enhance the prediction in the future, we plan to mine users' opinions, and contents consumed by users on social networks to implicitly extract their feedbacks and tastes. In addition, the recommendation quality could be improved by using some methods such as listwise, pairwise and pointwise ranking techniques to learn to accurately rank items recommended to users; and therefore refine the recommendation precision since users will receive a list of relevant and ranked items at the end of the process. We also plan to develop an improved data sampling strategy that selects only the most informative training instances to involve in the prediction process. Consequently, that sampling strategy will contribute to reduce the computational cost and speed up the recommendation process.

REFERENCES

- [1] H. Yan and Y. Tang, "Collaborative filtering based on Gaussian mixture model and improved Jaccard similarity," *IEEE Access*, vol. 7, pp. 118690–118701, 2019.
- [2] R. Logesh, V. Subramaniaswamy, V. Vijayakumar, and X. Li, "Efficient user profiling based intelligent travel recommender system for individual and group of users," *Mobile Netw. Appl.*, vol. 24, no. 3, pp. 1018–1033, Jun. 2019.
- [3] R. Chen, Q. Hua, Y.-S. Chang, B. Wang, L. Zhang, and X. Kong, "A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks," *IEEE Access*, vol. 6, pp. 64301–64320, 2018.
- [4] H. Parvin, P. Moradi, and S. Esmaeili, "TCFACO: Trust-aware collaborative filtering method based on ant colony optimization," *Expert Syst. Appl.*, vol. 118, pp. 152–168, Mar. 2019.
- [5] F. Zhao, Y. Shen, X. Gui, and H. Jin, "SDBPR: Social distance-aware Bayesian personalized ranking for recommendation," *Future Gener. Comput. Syst.*, vol. 95, pp. 372–381, Jun. 2019.
- [6] X. Chen, Y. Guo, Y. Yang, and Z. Mi, "Trust-based collaborative filtering algorithm in social network," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2016, pp. 1–5.
- [7] D. Kluver, M. D. Ekstrand, and J. A. Konstan, "Rating-based collaborative filtering: Algorithms and evaluation," in *Social Information Access: Systems and Technologies*, P. Brusilovsky and D. He, Eds. Cham, Switzerland: Springer, 2018, pp. 344–390.
- [8] Z. Yang, B. Wu, K. Zheng, X. Wang, and L. Lei, "A survey of collaborative filtering-based recommender systems for mobile Internet applications," *IEEE Access*, vol. 4, pp. 3273–3287, 2016.
- [9] X. Luo, Y. Xia, and Q. Zhu, "Applying the learning rate adaptation to the matrix factorization based collaborative filtering," *Knowl.-Based Syst.*, vol. 37, pp. 154–164, Jan. 2013.
- [10] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data," *Expert Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113248.

- [11] D. Lian, X. Xie, and E. Chen, "Discrete matrix factorization and extension for fast item recommendation," *IEEE Trans. Knowl. Data Eng.*, early access, Nov. 4, 2019, doi: [10.1109/TKDE.2019.2951386](https://doi.org/10.1109/TKDE.2019.2951386).
- [12] D. Li, C. Chen, Q. Lv, L. Shang, Y. Zhao, T. Lu, and N. Gu, "An algorithm for efficient privacy-preserving item-based collaborative filtering," *Future Gener. Comput. Syst.*, vol. 55, pp. 311–320, Feb. 2016.
- [13] H. Tian and P. Liang, "Personalized service recommendation based on trust relationship," *Sci. Program.*, vol. 2017, Mar. 2017, Art. no. 4106134.
- [14] Z. Tan and L. He, "An efficient similarity measure for user-based collaborative filtering recommender systems inspired by the physical resonance principle," *IEEE Access*, vol. 5, pp. 27211–27228, 2017.
- [15] X. Zheng, L. D. Xu, and S. Chai, "QoS recommendation in cloud services," *IEEE Access*, vol. 5, pp. 5171–5177, 2017.
- [16] K. Jayapriya, N. A. B. Mary, and R. S. Rajesh, "Cloud service recommendation based on a correlated QoS ranking prediction," *J. Netw. Syst. Manage.*, vol. 24, no. 4, pp. 916–943, Oct. 2016.
- [17] Z. Cui, X. Xu, F. Xue, X. Cai, Y. Cao, W. Zhang, and J. Chen, "Personalized recommendation system based on collaborative filtering for IoT scenarios," *IEEE Trans. Services Comput.*, early access, Jan. 7, 2020, doi: [10.1109/TSC.2020.2964552](https://doi.org/10.1109/TSC.2020.2964552).
- [18] Y. Du, N. Sutton-Charani, S. Ranwez, and V. Ranwez, "EBCR: Empirical Bayes concordance rate to weight similarity measurement within collaborative filtering recommendations," Ph.D. dissertation, Adv. Cognit. Autom., LGI2P/Ecole Mines d'Alès Alès, France, 2019, p. 12.
- [19] L. Ren and W. Wang, "An SVM-based collaborative filtering approach for top-N Web services recommendation," *Future Gener. Comput. Syst.*, vol. 78, pp. 531–543, Jan. 2018.
- [20] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowl.-Based Syst.*, vol. 97, pp. 188–202, Apr. 2016.
- [21] X. Ma, D. Guo, L. Cui, X. Li, X. Jiang, and X. Chen, "SOM clustering collaborative filtering algorithm based on singular value decomposition," in *Proc. 4th Int. Conf. Math. Artif. Intell. (ICMAI)*, 2019, pp. 61–65.
- [22] F. Ortega, A. Hernando, J. Bobadilla, and J. H. Kang, "Recommending items to group of users using matrix factorization based collaborative filtering," *Inf. Sci.*, vol. 345, pp. 313–324, Jun. 2016.
- [23] R. Lara-Cabrera, Á. González-Prieto, F. Ortega, and J. Bobadilla, "Evolving matrix-factorization-based collaborative filtering using genetic programming," *Appl. Sci.*, vol. 10, no. 2, p. 675, Jan. 2020.
- [24] K. Li, X. Zhou, F. Lin, W. Zeng, and G. Alterovitz, "Deep probabilistic matrix factorization framework for online collaborative filtering," *IEEE Access*, vol. 7, pp. 56117–56128, 2019.
- [25] H. Parvin, P. Moradi, S. Esmaili, and N. N. Qader, "A scalable and robust trust-based nonnegative matrix factorization recommender using the alternating direction method," *Knowl.-Based Syst.*, vol. 166, pp. 92–107, Feb. 2019.
- [26] R. Du, J. Lu, and H. Cai, "Double regularization matrix factorization recommendation algorithm," *IEEE Access*, vol. 7, pp. 139668–139677, 2019.
- [27] W. Zhou, J. Li, Y. Zhou, and M. H. Memon, "Bayesian pairwise learning to rank via one-class collaborative filtering," *Neurocomputing*, vol. 367, pp. 176–187, Nov. 2019.
- [28] X. Wu, Y. Fan, J. Zhang, H. Lin, and J. Zhang, "QF-RNN: QI-matrix factorization based RNN for time-aware service recommendation," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2019, pp. 202–209.
- [29] R. R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2007, pp. 1257–1264.
- [30] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 880–887.
- [31] G. Li, T. Zhu, J. Hua, T. Yuan, Z. Niu, T. Li, and H. Zhang, "Asking images: Hybrid recommendation system for tourist spots by hierarchical sampling statistics and multimodal visual Bayesian personalized ranking," *IEEE Access*, vol. 7, pp. 126539–126560, 2019.
- [32] S. Wen, C. Wang, H. Li, and G. Zheng, "Parallel naïve Bayes regression model-based collaborative filtering recommendation algorithm and its realisation on Hadoop for big data," *Int. J. Inf. Technol. Manage.*, vol. 18, nos. 2–3, pp. 129–142, 2019.
- [33] J. Yu, Z. Xuan, X. Feng, Q. Zou, and L. Wang, "A novel collaborative filtering model for LncRNA-disease association prediction based on the Naïve Bayesian classifier," *BMC Bioinf.*, vol. 20, no. 1, p. 396, Dec. 2019.
- [34] Y. Zhang, K. Meng, W. Kong, Z. Y. Dong, and F. Qian, "Bayesian hybrid collaborative filtering-based residential electricity plan recommender system," *IEEE Trans. Ind. Informat.*, vol. 15, no. 8, pp. 4731–4741, Aug. 2019.
- [35] P. Valdiviezo-Diaz, F. Ortega, E. Cobos, and R. Lara-Cabrera, "A collaborative filtering approach based on Naïve Bayes classifier," *IEEE Access*, vol. 7, pp. 108581–108592, 2019.
- [36] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales, "Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks," *Int. J. Approx. Reasoning*, vol. 51, no. 7, pp. 785–799, Sep. 2010.
- [37] X. Su and T. M. Khoshgoftar, "Collaborative filtering for multi-class data using Bayesian networks," *Int. J. Artif. Intell. Tools*, vol. 17, no. 1, pp. 71–85, Feb. 2008.
- [38] J. Pérez-Marcos, L. Martín-Gómez, D. M. Jiménez-Bravo, V. F. López, and M. N. Moreno-García, "Hybrid system for video game recommendation based on implicit ratings and social networks," *J. Ambient Intell. Hum. Comput.*, pp. 1–11, Jan. 2020.
- [39] Y. Tian, B. Zheng, Y. Wang, Y. Zhang, and Q. Wu, "College library personalized recommendation system based on hybrid recommendation algorithm," in *Proc. 11th CIRP Conf. Ind. Product-Service Syst.*, vol. 83, 2019, pp. 490–494.
- [40] R. Logesh and V. Subramaniaswamy, "Exploring hybrid recommender systems for personalized travel applications," in *Cognitive Informatics and Soft Computing*, P. K. Mallick, V. E. Balas, A. K. Bhoi, and A. F. Zobaa, Eds. Singapore: Springer, 2019, pp. 535–544.
- [41] H. Kung-Hsiang, F. Yi-Fu, L. Yi-Ting, L. Tzong-Hann, C. Yao-Chun, L. Yi-Hui, and L. Shou-De, "A-HA: A hybrid approach for hotel recommendation," in *Proc. Workshop ACM Rec. Syst. Challenge*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–5.
- [42] A. Gatzoura, J. Vinagre, A. M. Jorge, and M. Sánchez-Marrè, "A hybrid recommender system for improving automatic playlist continuation," *IEEE Trans. Knowl. Data Eng.*, early access, Nov. 8, 2019, doi: [10.1109/TKDE.2019.2952099](https://doi.org/10.1109/TKDE.2019.2952099).
- [43] P. Kouki, J. Schaffer, J. Pujara, J. O'Donovan, and L. Getoor, "Personalized explanations for hybrid recommender systems," in *Proc. 24th Int. Conf. Intell. User Interfaces (IUI)*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 379–390.
- [44] S. Paul and D. Das, "User-item-based hybrid recommendation system by employing mahout framework," in *Emerging Technology in Modelling and Graphics*, J. K. Mandal and D. Bhattacharya, Eds. Singapore: Springer, 2020, pp. 341–357.
- [45] A. Dhruv, A. Kamath, A. Powar, and K. Gaikwad, "Artist recommendation system using hybrid method: A novel approach," in *Emerging Research in Computing, Information, Communication and Applications*, N. R. Shetty, L. M. Patnaik, H. C. Nagaraj, P. N. Hamsavath, and N. Nalini, Eds. Singapore: Springer, 2019, pp. 527–542.
- [46] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward, "Better guarantees for k -means and euclidean k -median by primal-dual algorithms," *SIAM J. Comput.*, p. FOCS17–97, 2019.
- [47] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, and A. Prugel-Bennett, "Novel centroid selection approaches for KMeans-clustering based recommender systems," *Inf. Sci.*, vol. 320, pp. 156–189, Nov. 2015.
- [48] A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin, *Bayesian Data Analysis* (Chapman & Hall/CRC Texts in Statistical Science), 3rd ed. New York, NY, USA: Taylor & Francis, 2013.
- [49] L. D. Broemeling, *Bayesian Analysis of Time Series*. Boca Raton, FL, USA: CRC Press, 2019.
- [50] Y.-H. Chien and E. I. George, "A Bayesian model for collaborative filtering," in *Proc. AISTATS*, 2002, pp. 1–6.
- [51] D. Fink, *A Compendium of Conjugate Priors*, vol. 46. Citeseer, 1997. [Online]. Available: <http://www.people.cornell.edu/pages/df36/CONJINTRnew%20TEX.pdf>
- [52] D. van Ravenzwaaij, P. Cassey, and S. D. Brown, "A simple introduction to Markov Chain Monte-Carlo sampling," *Psychonomic Bull. Rev.*, vol. 25, pp. 143–154, Feb. 2018.
- [53] S. Herholz, Y. Zhao, O. Elek, D. Nowrouzehzrai, H. P. A. Lensch, and J. Křivánek, "Volume path guiding based on zero-variance random walk theory," *ACM Trans. Graph.*, vol. 38, no. 3, pp. 1–19, Jun. 2019.
- [54] F. Ortega, B. Zhu, J. Bobadilla, and A. Hernando, "CF4J: Collaborative filtering for java," *Knowl.-Based Syst.*, vol. 152, pp. 94–99, Jul. 2018.



ARMIELLE NOULAPEU NGAFFO (Graduate Student Member, IEEE) received the master's degree in telecommunication services engineering from the National Advanced School of Posts, Telecommunications and ICT (SUP'PTIC), Yaoundé, Cameroon, in 2016. She is currently pursuing the Ph.D. degree with the Higher School of Communications of Tunis (SUP'COM). She is also a member of the Mediatron Laboratory.

Her research interests are recommender systems, sentiment analysis, artificial intelligence, services discovery/exposure, big data analysis, cloud computing, virtualization, service-oriented architecture, and services engineering.



WALID EL AYE B received the degree in software engineering from the National Institute of Applied Sciences and Technology (INSAT), in 2011, and the Ph.D. degree from the Higher School of Communications of Tunis (SUP'COM), University of Carthage, in 2017. He is currently a Teacher and a Researcher with the Mediatron Laboratory. His research interests include services engineering, distributed systems, quality of service, network security, service-oriented architecture, big data,

cloud computing, fuzzy systems, optimization techniques, service discovery/exposure, the Internet of Things, 4G mobile communication, graph theory, 5G mobile communication, and IP multimedia subsystems.



ZIÈD CHOUKAIR received the Ph.D. degree from the University of Paris-Sud XI. He was an Associate Professor of computer science with the National School of Telecommunications of Brittany (ENST), in 2003. He was an Associate Professor of computer science and a Researcher with the IT Research Institute of Toulouse, in 2005. He is currently a Full Professor with the Higher School of Communications of Tunis (SUP'COM), University of Carthage. He is also with the Mediatron

Laboratory. His research interests include services engineering, service-oriented architectures, telecommunication service platforms, service discovery/exposure, machine-to-machine communication, distributed systems, big data, cloud computing, graph theory, artificial intelligence, multi-access system, 3G mobile communication, IT service management, virtualization, migration to the cloud, the Internet of Things, mobile communication systems, IP networks, fuzzy set theory, 4G mobile communication, 5G mobile communication, and IP multimedia subsystems.

...