



Service recommendation driven by a matrix factorization model and time series forecasting

Armienne Noulapeu Ngaffo¹ · Walid El Ayeb¹ · Zied Choukair¹

Accepted: 27 April 2021 / Published online: 16 May 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The rise of high-quality cloud services has made service recommendation a crucial research question. Quality of Service (QoS) is widely adopted to characterize the performance of services invoked by users. For this purpose, the QoS prediction of services constitutes a decisive tool to allow end-users to optimally choose high-quality cloud services aligned with their needs. The fact is that users only consume a few of the broad range of existing services. Thereby, perform a high-accurate service recommendation becomes a challenging task. To tackle the aforementioned challenges, we propose a data sparsity resilient service recommendation approach that aims to predict relevant services in a sustainable manner for end-users. Indeed, our method performs both a QoS prediction of the current time interval using a flexible matrix factorization technique and a QoS prediction of the future time interval using a time series forecasting method based on an AutoRegressive Integrated Moving Average (ARIMA) model. The service recommendation in our approach is based on a couple of criteria ensuring in a lasting way, the appropriateness of the services returned to the active user. The experiments are conducted on a real-world dataset and demonstrate the effectiveness of our method compared to the competing recommendation methods.

Keywords Matrix factorization · QoS prediction · AutoRegressive integrated moving average model · Service recommendation · Time series forecasting

1 Introduction

The rapid development of broadband access technologies has led to an escalation of rich and varied cloud services and applications. The plethora of existing services complicates decision-making for internet users. Recommender systems address this issue by proposing algorithms to predict users' sensitivity for services aligned with their needs. According to the literature review, recommender systems are popularly implemented through Collaborative Filtering (CF) [22].

Collaborative-filtering-based recommender systems are mainly organized into two types, namely memory-based CFs and model-based CFs. Among memory-based CFs, we identify user-based CFs [11, 12] and item-based CFs [14] whose respective approaches are based on the study of interactions between users and items to assess inter-user similarities and inter-item similarities. Memory-based CFs are based on the assumption that, within a group of users with similar or identical behaviors, the previous experiences of certain users with regard to consumed services, can help to predict those of others relative to the same services. Memory-based CFs are easy to implement and present an acceptable quality of prediction; however, they are based on resource-intensive algorithms due to the fact that computational calculations are applied to the full data matrix [24]. In other words, memory-based CFs display scalability limitations, an inappropriateness for large datasets, poor performances in case of data sparseness and during the cold start problem for new users and new items [2]. Model-based CFs overcome the limitations displayed by memory-based CFs. Indeed, model-based CFs present the advantage of their effectiveness in the case of large datasets and data sparsity. Among model-based CFs,

✉ Armienne Noulapeu Ngaffo
armienne.ngaffo@supcom.tn

Walid El Ayeb
walidelayeb@supcom.tn

Zied Choukair
choukair@supcom.tn

¹ Higher School of Communication of Tunis,
Mediatron Laboratory, University of Carthage, Tunis, Tunisia

we identify those based on matrix factorization techniques, those based on clustering and others based on mathematical models such as Bayesian estimate, deep learning, random walker, etc [30].

The static nature of the data is an assumption on which classic recommender systems are modeled. However, a realistic recommendation approach should take into account the variability of users' behavior over time [4]. To address this issue, time-aware recommenders systems have been designed with the objective of integrating the dynamicity of users' needs and behaviors over time [21]. Existing time-aware recommendation solutions only predict the current moment and display poor performances in the case of data sparsity. In reality, faced with a multitude of services, users invoke and co-invoke only a small group of services, thus causing increased porosity of the data matrix with a large quantity of missing data. The aforementioned inadequacies of the state-of-the-art recommendation methods constitute our motivation to propose a data sparsity resilient and high-accurate recommendation approach that aims to predict the current needs but also the near future needs of end-users. We propose a service recommendation approach based on a latent feature unsupervised learning coupled with a time series forecasting method. In this paper, our contribution is highlighted through the following aspects:

- By applying a flexible matrix factorization technique, our method addresses the data sparsity problem in the presence of a large amount of missing data.
- Our approach is based on time series forecasting method for the prediction of the future time slot t_{c+1} using the prediction of the current time slot t_c . The prediction of the future time slot anticipates the near future needs of the active user.
- The variability of users' behavior and the impact of data recentness are integrated into our approach by the definition of a decay function allowing to sketch the evolution of user interest profiles over time.
- The services recommended to the active user meet a couple of criteria (q_{c_0}, q_{c_0+1}); thus ensuring the exclusive selection of services whose prediction of the current time t_c , and that of the future time slot t_{c+1} are among the highest.
- Extensive experiments are conducted on real-world datasets and highlight the effectiveness of our proposal compared to state-of-the-art recommendation methods.

The remainder of this paper is structured as follows: Section 2 presents state-of-the-art recommendation methods, Section 3 presents our recommendation approach. Section 4 describes the experiments and the results obtained.

Section 5 concludes this paper and presents our perspectives.

2 Related work

The growth of the number of cloud services and the difficulty for internet users to choose services matching their needs have motivated scientific research in the field of recommender systems. In this section, we survey the state-of-the-art methods related to recommender systems.

2.1 Memory-based collaborative filtering

Memory-based CFs present the advantage of their explainability and easy-implementation. They are based on the assumption that within a group of users with identical or similar behaviors, the previous experiences of certain users can be used to predict those of others with regard to the services never invoked. Within memory-based CFs, a distinction is made between user-based CFs, item-based CFs and user-item-based CFs [30]. A cloud service recommendation based on user-based CF is proposed by the authors [10, 25]. The authors assess inter-user similarities using the Spearman correlation coefficient [6]. While the authors [32] propose a Cloud service recommendation method based on the calculation of inter-user similarities using the Kendall rank correlation coefficient [20]. The researchers [14] offer an efficient privacy-preserving recommendation method based on an item-based CF. The authors assess the inter-item similarities using a modified Pearson measure to preserve user privacy. In [31], the authors offer a web service recommendation approach based on hybrid user-item-based CFs. Using Pearson correlation measure [12], the authors assess inter-user similarities and inter-service similarities to reconcile the advantages of user-based CFs and item-based CFs.

The aforementioned recommendation methods are greedy in terms of computational resources, non-scalable and low-accurate. Indeed, similarity computations on large datasets are hard-achievable. Furthermore, these memory-based recommendation approaches display poor performances in the case of data sparseness; meaning in the presence of a large amount of missing data. To overcome these shortcomings, model-based recommender systems are widely adopted for their effectiveness.

2.2 Model-based collaborative filtering

Model-based CFs overcome the limitations displayed by memory-based CFs. Indeed, model-based CFs are highly

accurate, scalable and suitable for large datasets [2]. In the literature review, we distinguish the model-based CF implemented based on the matrix factorization technique, those whose implementation is based on clustering-based algorithms and others based on deep learning, the bayesian network, etc. Yang et al [30]. The authors [16] offer a recommendation approach based on both user-based CFs and item-based CFs. The method proposed by the authors is based on the Bayesian estimate of the probability with which the active user rates an item. The authors' method reconciles the efficiency of model-based CFs and the easy-understandability of memory-based CFs. Researchers [33] offer a cold start recommendation method. To alleviate the cold start problem for new elements in the recommender system, the authors adopt the matrix factorization technique by reducing the original high-rank matrix to low-rank matrices. In [8], the authors predict users' tastes by applying matrix factorization based on a Bayesian probabilistic model. The authors' proposal in [13] performs an improved matrix factorization technique for predicting metabolite-disease associations. In [15, 23], the service recommendation is based on a matrix factorization supported by deep learning algorithms. The model-based CFs are scalable, robust and efficient even in case of data sparsity, case of a cold start problem for newly added users and items; they are suitable for large datasets. However, the above-mentioned methods do not take into account the dynamic nature of users' needs over time.

2.3 Time-aware recommender systems

Classic recommender systems assume static users' behavior over time. In reality, user preferences display variability over time which is ignored by existing recommenders systems. To address this issue, time-aware recommender systems [17, 18, 29] have been developed to integrate the dynamic nature of the service performances. In [28], the authors offer a time-aware service recommendation approach based on the tensor factorization technique. The authors perform a personalized QoS prediction based on a latent feature learning. However, the method proposed by the authors does not integrate the variability of users' needs over time. In other words, the data recentness is ignored by the authors' approach; all data whatever their seniority, contribute identically in the prediction process. Following [5], they propose a cloud service recommendation approach based on user-based CF and an ARIMA model. Their method focused on the inter-user similarity evaluation displays limitations faced with the cold start problem and data sparsity. The authors in [7] propose a forecasting ARIMA-based model to predict the Coronavirus evolution

throughout the world. Of course, in the literature, some time-aware recommendation approaches to consider the users' variability over time, use a decay function. Most often, the decay function is coupled to memory-based approaches [5, 19, 26]. The matrix factorization technique is popularly adopted in several time-aware recommender systems thanks to its efficiency. However, a properly chosen decay function could help to accurately model the dynamic character of users' tastes over time. In addition, the baseline matrix factorization technique used in the state-of-the-art time-aware recommender systems can be also improved to accurately model intricate user-items interactions.

The novelty of our proposal relies on a fitting proposed decay function embedded into a flexible matrix factorization process. The proposed decay function properly sketches the variability of users' needs over time and therefore, contributes to the refinement of users' tastes prediction. Moreover, the used matrix factorization technique is doubly biased to accurately model intricate user-service interactions. In addition, ARIMA models are most often used for memory-based recommender systems that are very poor-scalable. To remedy this limitation, our proposal uses the effectiveness of ARIMA model combined with the robustness of model-based recommender systems such as matrix factorization-based ones.

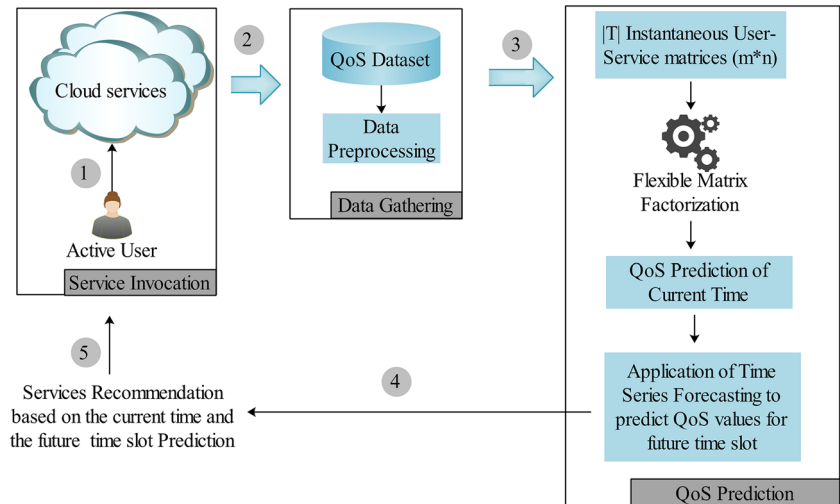
In this paper, we address the data sparsity problem and the problem of service targeting. We are interested in the proposal of a service recommendation model which is data sparseness robust and which considers the dynamic nature of needs and the behavior of users over time. The approach presented in this article aims to predict the current moment and also the future time slot in order to anticipate the near future users' needs. Our recommendation method is based on two main prediction phases. The first step is to predict the current time interval. At this phase, a flexible matrix factorization technique is applied to a set of instantaneous data matrices in order to perform the QoS prediction for the current time slot. The second step is devoted to predicting future time intervals using a time series forecasting method based on an AutoRegressive Integrated Moving Average (ARIMA) model.

The next section presents our TASERM approach.

3 Time series analysis based service recommendation using matrix factorization (TASERM)

Our approach aims to make a QoS prediction of the current time interval and the future time slot based on past historical

Fig. 1 TASERM Model



QoS experiences. The purpose is to return to the active user, services which meet his current and near future needs. In this way, the relevance of the results returned to the end-user is extended over time in order to durably meet the current and future expectations of the active user. Figure 1 represents our TASERM’s model workflow. It works as follows:

- The active user invokes cloud services thus generating QoS data which are stored in the database. These QoS data are derived from service invocations spread over regular time slots.
- During the data preprocessing, QoS data are normalized and a logistic function is applied on collected data to consider the dynamic character of users’ behaviors over time.
- Thereafter, for each time slot, a user-service matrix is constructed. Therefore, the $|\Psi|$ instantaneous user-service matrices are obtained knowing that $|\Psi|$ represents the number of time slots in the observation window.
- The QoS prediction is carried out by a flexible matrix factorization technique in order to predict the current time slot t_c .
- The future time slot t_{c+1} is predicted using a time series forecasting method based on an ARIMA model.
- Services whose predicted QoS data at the current time t_c and future time slot t_{c+1} are the highest, are recommended to the active user.

The next subsection presents the mathematical formulation of our problem.

3.1 Problem formulation

We consider a recommender system in which m users belonging to the set $U = \{u_1, u_2, \dots, u_i, \dots, u_m\}$ invoke n services belonging to the set $S = \{s_1, s_2, \dots, s_j, \dots, s_n\}$ during a time interval t_k , where $t_k \in \Psi = \{t_1, t_2, \dots, t_k, \dots, t_c\}$ (see Fig. 2). At each invocation of a service s by a user u during a time slot t_k , the observed QoS value is recorded as $q_u^s(t_k)$ in the instantaneous user-service matrix $M_k = [q_u^s(t_k)]_{m \times n}$ where $u \in U, s \in S, t_k \in \Psi$. Deductively, on the observation window Ψ , the QoS data are recorded in the set $M = \{M_1, M_2, \dots, M_k, \dots, M_c\}$ of the instantaneous user-service matrices respectively collected during instants $t_1, t_2, \dots, t_k, \dots, t_c$.

A logistic function is applied on collected data to consider the data recentness impact. Indeed, the more recent

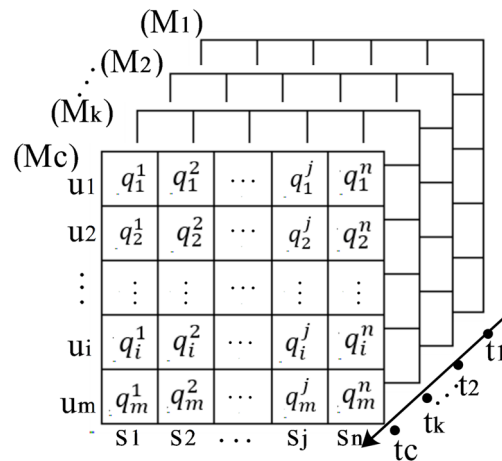


Fig. 2 Our mathematical model

the QoS values involved in the system, the more they contribute significantly to the prediction process [26]. Our decay function is defined by:

$$y(t) = \exp(-\gamma|t - t_c|), \tag{1}$$

where $t \in \Psi$ and γ is the decay rate. An increase of the decay rate induces a faster decrease of the decay function.

We perform a rescaling to [1, 5] in order to have a similarity to ratings that are widely used to feed recommender systems. The values obtained by the prediction process can be transformed to their original status by using the reverse scaling formula. Without loss of generality, QoS values are rescaled to the interval [1, 5] using the following function:

$$g(x) = \begin{cases} 1, & \text{if } x < Q_{\min} \\ 5, & \text{if } x > Q_{\max} \\ 1 + \frac{4(x - Q_{\min})}{Q_{\max} - Q_{\min}}, & \text{otherwise} \end{cases}, \tag{2}$$

where Q_{\min} and Q_{\max} are respective minimal and maximal bounds of QoS values.

The predicted score after the scaling step could not be the same since the scaling process is directly applied to QoS values that are thereafter embedded in the prediction formula.

3.2 QoS prediction for current time slot

In this subsection, the goal is to predict the current time slot t_c . For this purpose, we are interested in the current user-service matrix $M_c = [q_u^s(t_c)]_{m \times n}$. The matrix factorization technique is a method widely adopted in recommender systems [2]. It is based on unsupervised feature learning and consists of reducing the computational load induced by the prediction process. By applying matrix factorization, the original high-dimensional matrix M_c (simply noted in M thereafter) is approximated by two low-rank matrices (see Fig.3).

To predict the QoS value of the current time interval, we adopt a progressive reasoning based on the following assumptions:

Assumption 1 *QoS data are assumed to be static over time.*

According to the matrix factorization, the QoS value q_u^s is approximated as follows:

$$q_u^s \approx \sum_{r \in F} w_{ur} v_{sr}, \tag{3}$$

where $r \ll \min(m, n)$ is the number of latent features and $r \in F = \{1; 2; \dots; f - 1; f\}$. The product $w_{ur} v_{sr}$ approximatively assesses the interaction between a user u and a service s .

From the matrix point of view, the approximation of the user-service matrix $M = [q_u^s]_{m \times n}$ by the matrix \hat{M} is defined as follows:

$$\hat{M} = WV = [\hat{q}_u^s]_{m \times n}, \tag{4}$$

where $W = [w_{ur}]_{m \times f}$ and $V = [v_{sr}]_{f \times n}$ are the two low-rank matrices from factorization process. The user latent factor matrix W means the users' interest for each latent factor while V evaluates the interest aroused by each latent factor.

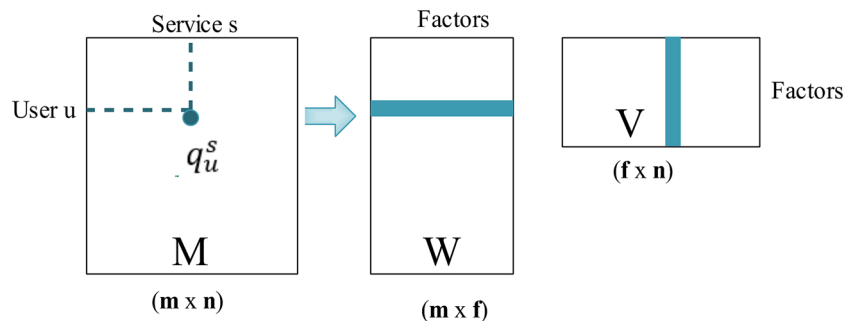
We consider the bias b_{us} to compensate for the QoS values variations induced by the interactions between a user u and a service s as follows [27]:

$$b_{us} = \bar{q} + b_u + b_s, \tag{5}$$

where b_u is the difference between the overall QoS values average \bar{q} and the average of QoS values from u 's service invocations, b_s is the difference between the overall QoS values average \bar{q} and the average of QoS values related to s service invocations. The overall QoS values average \bar{q} is computed as follows:

$$\bar{q} = \frac{\sum_{u \in U} \sum_{s \in S} I_{us} q_u^s}{\sum_{u \in U} \sum_{s \in S} I_{us}}, \tag{6}$$

Fig. 3 Matrix factorization



where I_{us} is the indicator function which is equal to 1 when a user u invokes a service s and 0 otherwise.

By applying the bias b_{us} , the approximation of the QoS value q_u^s is reformulated as follows:

$$q_u^s \simeq b_{us} + \sum_{r \in F} w_{ur} v_{sr} = \widehat{q}_u^s. \tag{7}$$

From the matrix point of view, the approximative user-service matrix \widehat{M} is defined as follows:

$$\widehat{M} = B + WV, \tag{8}$$

where $B = [b_{us}]_{m \times n}$ is the bias matrix with $u \in U, s \in S$.

The matter now is to minimize the approximation error E defined as follows :

$$E = \frac{1}{2} \|M - \widehat{M}\|_F^2, \tag{9}$$

where is $\|\cdot\|_F^2$ the Frobenius norm [1].

The minimization of the approximation error of the matrix M by the matrix \widehat{M} is an optimization problem whose objective function is defined as follows:

$$\min_{W^{opt}, V^{opt}} E = \frac{1}{2} \|M - \widehat{M}\|_F^2 = \frac{1}{2} \sum_{u \in U} \sum_{s \in S} I_{us} (q_u^s - \widehat{q}_u^s)^2, \tag{10}$$

where $\widehat{q}_u^s = \bar{q} + b_u + b_s + \sum_{r \in F} w_{ur} \cdot v_{sr}$.

To avoid overfitting problem, additional parameter ρ is integrated in the objective function expression as follows:

$$\begin{aligned} & \min_{W^{opt}, V^{opt}} \frac{1}{2} \sum_{u \in U} \sum_{s \in S} I_{us} \left(\left(q_u^s - b_{us} - \sum_{r \in F} w_{ur} v_{sr} \right)^2 \right. \\ & \left. + \rho \left(\sum_{u \in U} \|w_u\|^2 + \sum_{s \in S} \|v_s\|^2 + \sum_{u \in U} b_u^2 + \sum_{s \in S} b_s^2 \right) \right) \\ & \Leftrightarrow \min_{W^{opt}, V^{opt}} \frac{1}{2} \sum_{u \in U} \sum_{s \in S} I_{us} \left(\left(q_u^s - \bar{q} - b_u - b_s - \sum_{r \in F} w_{ur} v_{sr} \right)^2 \right. \\ & \left. + \rho \left(\sum_{u \in U} \sum_{r \in F} w_{ur}^2 + \sum_{s \in S} \sum_{r \in F} v_{sr}^2 + \sum_{u \in U} b_u^2 + \sum_{s \in S} b_s^2 \right) \right) \end{aligned} \tag{11}$$

Assumption 2 QoS data are assumed to be dynamic over time.

We consider the variability of QoS values over time and the data recentness impact; the objective function is reformulated as follows:

$$\begin{aligned} & \min_{W^{opt}, V^{opt}} \frac{1}{2} \sum_{u \in U} \sum_{s \in S} y(t) \cdot I_{us}(t) \left((q_u^s(t) - \widehat{q}_u^s(t))^2 \right) \\ & + \rho \left(\sum_{u \in U} \sum_{r \in F} w_{ur}^2(t) + \sum_{s \in S} \sum_{r \in F} v_{sr}^2(t) + \sum_{u \in U} b_u^2(t) + \sum_{s \in S} b_s^2(t) \right) \\ & \Leftrightarrow \min_{W^{opt}, V^{opt}} \frac{1}{2} \sum_{u \in U} \sum_{s \in S} \exp(-\gamma|t - t_c|) \cdot I_{us}(t) \left((q_u^s(t) - \widehat{q}_u^s(t))^2 \right) \\ & + \rho \left(\sum_{u \in U} \sum_{r \in F} w_{ur}^2(t) + \sum_{s \in S} \sum_{r \in F} v_{sr}^2(t) + \sum_{u \in U} b_u^2(t) + \sum_{s \in S} b_s^2(t) \right), \end{aligned} \tag{12}$$

where $\widehat{q}_u^s(t) = \bar{q}(t) + b_u(t) + b_s(t) + \sum_{r \in F} w_{ur}(t) \cdot v_{sr}(t)$.

The challenge now is to solve the aforementioned optimization problem. The stochastic gradient descent is a widely used as an optimization method [30]. We adopt this method to solve our optimization problem. The stochastic gradient descent is based on an iterative algorithm which updates W and V matrices following the direction of the gradient descent of the objective function. The update rules for low-rank matrices W and V are defined by (13) and (14).

$$\begin{aligned} \frac{\partial D}{\partial W} &= \sum_{t \in \Psi} \sum_{u \in U} \sum_{s \in S} y(t) \cdot I_{us}(t) \left(\left(\widehat{q}_u^s(t) - q_u^s(t) \right) \sum_{r \in F} v_{sr}(t) \right. \\ & \left. + \rho \sum_{u \in U} \sum_{r \in F} w_{ur}(t) \right). \end{aligned} \tag{13}$$

$$\begin{aligned} \frac{\partial D}{\partial V} &= \sum_{t \in \Psi} \sum_{u \in U} \sum_{s \in S} y(t) \cdot I_{us}(t) \left(\left(\widehat{q}_u^s(t) - q_u^s(t) \right) \sum_{r \in F} w_{ur}(t) \right. \\ & \left. + \rho \sum_{s \in S} \sum_{r \in F} v_{sr}(t) \right). \end{aligned} \tag{14}$$

The stochastic gradient descent process starts with the initialization of the latent features matrices W and V with random positive values. The optimal matrices W^{opt} and V^{opt} which minimize the objective function are obtained during the convergence of iterations carried out based on (13) and (14). Algorithm 1 describes the latent factors unsupervised learning process.

Algorithm 1 Latent factor unsupervised learning.**Data:**

- M : Set of $|\Psi|$ instantaneous user-service matrices
- $y(t)$: Decay function
- Ψ : Observation window
- ρ : Parameter to avoid overfitting problem
- f : Number of latent factors
- W : User latent factor matrix
- V : Service latent factor matrix
- S : Set of services
- U : Set of users

Result: Updated W and V

```

1 Begin
2 for ( $t \in \Psi$ ) do
3   for ( $r \in F$ ) do
4     for ( $u \in U$ ) do
5       Initialize  $w_{ur}(t)$  with random values;
6     end
7     for ( $s \in S$ ) do
8       Initialize  $v_{sr}(t)$  with random values;
9     end
10  end
11 end
12 for ( $t \in \Psi$ ) do
13    $q(t) = \frac{\sum_{u \in U} \sum_{s \in S} I_{us}(t) q_u^s(t)}{I_{us}(t)}$ 
14 end
15 repeat
16   for ( $t \in \Psi$ ) do
17     for ( $u \in U$ ) do
18       for ( $s \in S$ ) do
19          $\hat{q}_u^s(t) = q(t) + b_u(t) + b_s(t) + \sum_{r \in F} w_{ur}(t) \cdot v_{sr}(t)$ ;
20       end
21     end
22   end
23   for ( $t \in \Psi$ ) do
24     for ( $u \in U$ ) do
25       for ( $s \in S$ ) do
26         for ( $r \in F$ ) do
27            $w_{ur}(t) \leftarrow w_{ur}(t) - (\hat{q}_u^s(t) - q_u^s(t)) \cdot v_{sr}(t) \cdot y(t) - \rho \cdot w_{ur}(t) \cdot y(t)$ ;
28            $v_{sr}(t) \leftarrow v_{sr}(t) - (\hat{q}_u^s(t) - q_u^s(t)) \cdot w_{ur}(t) \cdot y(t) - \rho \cdot v_{sr}(t) \cdot y(t)$ ;
29         end
30       end
31     end
32   end
33 until Convergence;

```

In the next subsection, we perform the QoS Prediction for the future time slot.

3.3 QoS prediction for future time slot

AutoRegressive Integrated Moving Average (ARIMA) models are widely adopted for time series forecasting [5]. Based on the Box-Jenkins approach [3], ARIMA models predict future values of time series by extrapolating past values. We perform the QoS prediction for the future time slot based on the ARIMA-based prediction method in [9]. The predicted value $q_u^s(t_{c+1})$ of the future time interval is computed as follows:

$$q_u^s(t_{c+1}) = \sum_{t=t_{c-a+1}}^{t_c} \phi_t \cdot q_u^s(t) + \sum_{t=t_{c-b+1}}^{t_c} \theta_b \cdot \varepsilon(t) + \varepsilon(t_c), \quad (15)$$

where a and b are the orders of ARMA model; $q_u^s(t_{c+1})$ is the QoS value recorded from s service invocation by u user at the future time slot t_{c+1} . Parameters $\varepsilon(t)$ with $t = t_{c-b+1}, \dots, t_c$ correspond to past and current errors independently and normally distributed following the Gaussian probability density with a mean set to 0 and variance σ^2 . Parameters ϕ_t and θ_b are respectively the autoregressive coefficient and the moving average coefficient. They correspond to parameters of Maximum Likelihood Estimate (MLE), which maximize the likelihood function η defined as follows:

$$\infty (\sigma^2)^{-\frac{|\Psi|}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{t \in \Psi} \varepsilon(t)^2\right), \quad (16)$$

where $|\Psi|$ is the number of time slots in the observation window Ψ . To integrate the impact of the data recentness, we apply the decay function effect on the QoS prediction of the future time slot. For this purpose, we customize the estimate of QoS value $q_u^s(t_{c+1})$ as follows:

$$q_u^s(t_{c+1}) = \sum_{t=t_{c-a+1}}^{t_c} \phi_t \cdot y(t) q_u^s(t) + \sum_{t=t_{c-b+1}}^{t_c} \theta_b \cdot \varepsilon(t) + \varepsilon(t_c), \quad (17)$$

where $y(t)$ is the decay function.

The ARIMA based prediction is presented through a user u regarding a service s in order to deeply appreciate how the forecasting process has been performed. Thereafter, the forecasting can be extended to the whole set of users and the set of existing services.

The next subsection describes the service recommendation.

3.4 Service recommendation

The service recommendation is based on the coupling of the QoS prediction of the current time interval and the future time interval. Indeed, our method aims to guarantee the appropriateness of the services offered at the current time slot; but our approach also aims to ensure that the relevant services for the active user at the current time, that these services remain relevant in a lasting manner on the next time interval after the current time. Therefore, predicting only the current time may not be sufficient to meet the active user’s expectations in a sustainable manner. To alleviate this challenge, the prediction of the future time interval aims to anticipate the future needs of the active user with the purpose of extending over time the active user’s satisfaction.

The service selection lays on criteria (q_{c_0}, q_{c_0+1}) describing the set of recommended services $S_{rec} = \{s \in S | q_u^s(t_c) \geq q_{c_0} \wedge q_u^s(t_{c+1}) \geq q_{c_0+1}\}$, where q_{c_0} and q_{c_0+1} are thresholds in order to solely select high-quality services. $q_u^s(t_c)$ and $q_u^s(t_{c+1})$ are respectively QoS values from s service invocation by u user for the current time interval and the future time interval. By including the aforementioned steps of the prediction process, Algorithm 2 describes the service recommendation process.

Algorithm 2 Service recommendation.

Data:

- M : Set of $|\Psi|$ instantaneous user-service matrices
- $y(t)$: Decay function
- Ψ : Observation window
- u : The active user profile
- N : The number of recommended services
- q_{c_0}, q_{c_0+1} : Prediction thresholds
- S : Set of services
- U : Set of users

Result: S_{rec} Set of Top-N recommended services

```

1 Begin
2 for ( $s \in S$ ) do
3   | Predict  $q_u^s(t_c)$  using Algorithm 1;
4 end
5 for ( $s \in S$ ) do
6   |  $q_u^s(t_{c+1}) =$ 
       |  $\sum_{t=t_{c-a+1}}^{t_c} \phi_t \cdot y(t)q_u^s(t) + \sum_{t=t_{c-b+1}}^{t_c} \theta_b \cdot \varepsilon(t) + \varepsilon(t_c);$ 
7   | if  $q_u^s(t_c) \geq q_{c_0} \wedge q_u^s(t_{c+1}) \geq q_{c_0+1}$  then
8     | add service  $s$  to  $S_{rec}$ ;
9   | end
10 end
11 Sort  $S_{rec}$  by decreasing order;
12 Return Top-N of  $S_{rec}$ ;

```

The next section presents the performed experiments and results.

4 Experiments and results

In this section, the performances of our method are evaluated comparatively to existing recommender systems. Experiments are performed on a real-world dataset of QoS values from web service invocations. Thereafter, TASERM’s performances are studied comparatively to other recommendation methods.

The next subsection describes the experiment process.

4.1 Experiments setup

We have performed experiments on a computer with a processor of type Intel Core i7 (2.4 GHz) with 16 GB RAM, running Windows 10 Operating System. The free-available Anaconda Distribution version 1.8.5 has been used to implement our algorithm. From Anaconda, we have used Spyder version 3.2.6 as Python environment development.

The experiments have been performed using an open real-world dataset of QoS values from web services invocations. The researchers [28] have collected QoS values using WSMonitor tool to record performances during web services invocations by distributed computers from PlanetLab (<https://www.planet-lab.org/>). In this dataset, we use the response time and throughput performances from invocations of 4500 web services by 142 users during 64 regular time slots. Each time slot lasts 15 minutes. Thereafter, 64 instantaneous user-service matrices of dimension $142 * 4500$ have been built.

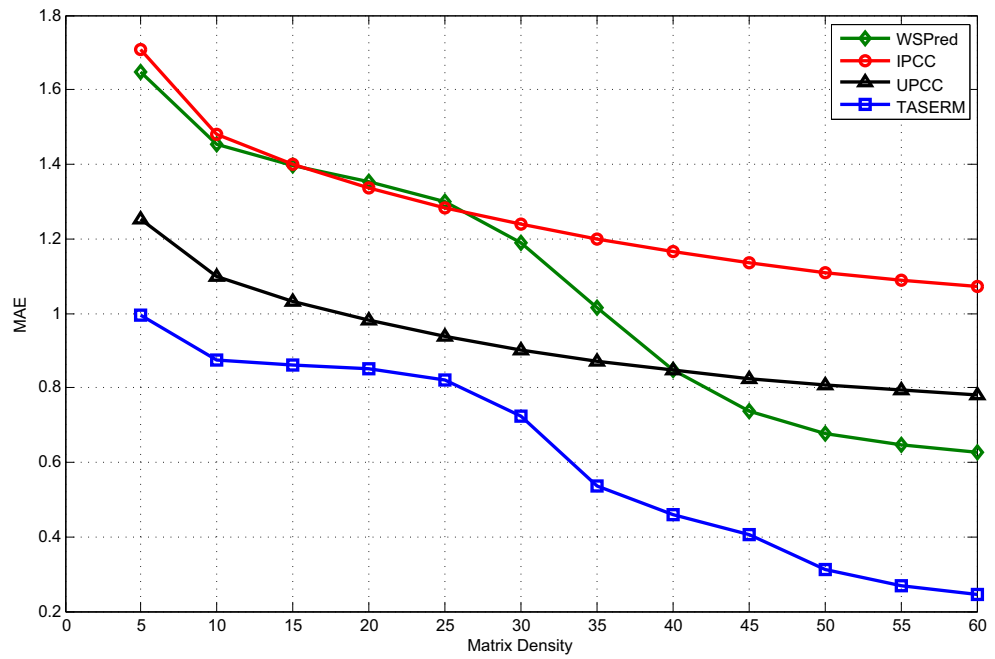
The matrix from the 63rd time slot represents the current time matrix; while the matrix from the 64th time slot represents the future time matrix. The current time matrix hosts QoS values to split into test data and training data. We progressively dig the current time matrix to sketch the data sparsity effect. For this purpose, the matrix density is defined as the matrix porosity intensity; meaning the training data percentage ranging from 5% to 60% in steps of 5%. For instance, a matrix density set to 10% means that 10% of training data and 90% of test data are represented in the current time matrix. Moreover, we blend 85% of training data hosted in the current time matrix with QoS values issued from other matrices.

The next subsection presents the evaluation metrics used to study TASERM’s performances.

4.2 Evaluation metrics

The TASERM’s performances are measured using key indicators namely the Mean Absolute Error (MAE), the

Fig. 4 Matrix Density Impact on MAE Performances with $r = 20$ (throughput)



Root Square Mean Error (RSME) and the Normalized Discounted Cumulative Gain (NDCG). MAE and RSME indicators evaluate the prediction accuracy. The lower values of MAE and RSME, the higher prediction accuracy. MAE and RSME values are computed as follows:

$$MAE = \frac{\sum_{s \in S_{rec}} |q_u^s(t) - \hat{q}_u^s(t)|}{N}, \tag{18}$$

$$RMSE = \sqrt{\frac{\sum_{s \in S_{rec}} (q_u^s(t) - \hat{q}_u^s(t))^2}{N}}, \tag{19}$$

where $q_u^s(t)$ is the original QoS value and $\hat{q}_u^s(t)$ the approximated value; N is the number of recommended services.

NDCG is widely adopted to evaluate the service ranking accuracy and is computed as follows:

Fig. 5 Matrix Density Impact on RSME Performances with $r = 20$ (throughput)

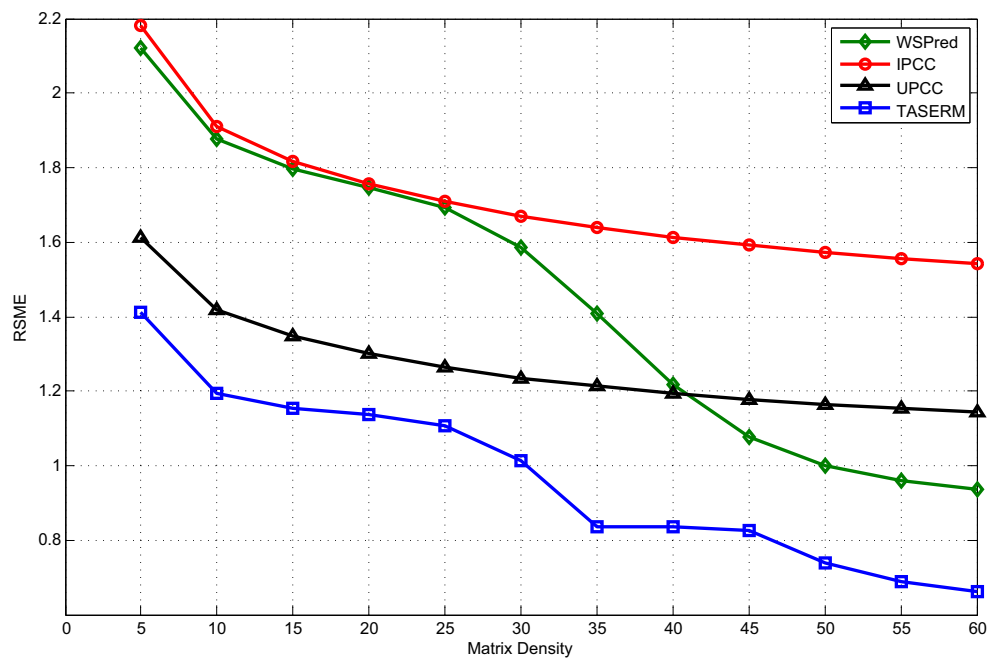
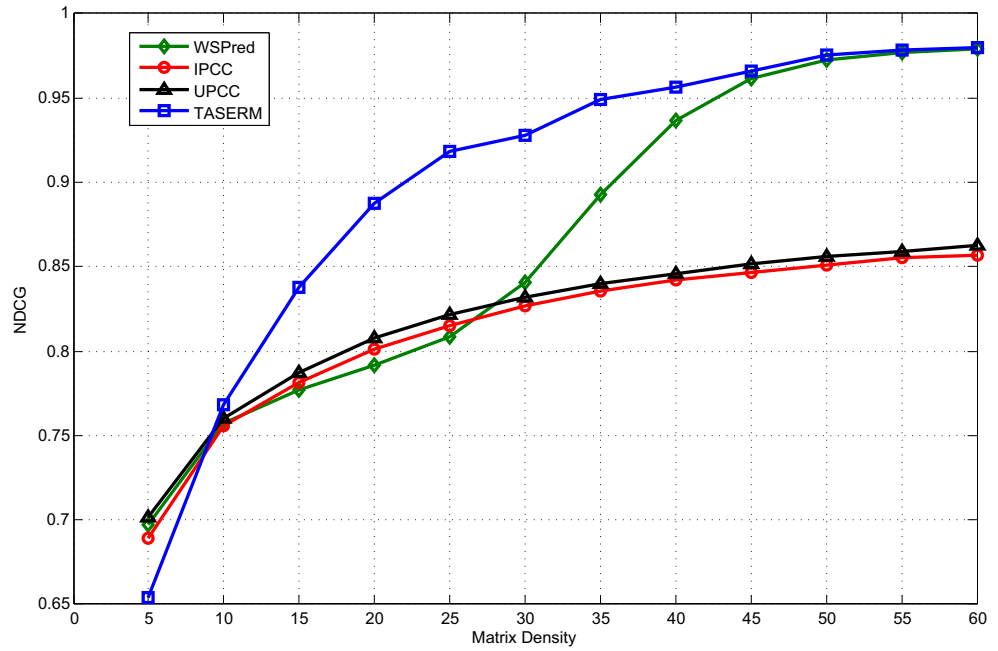


Fig. 6 Matrix Density Impact on NDCG Performances with $r = 20$ (throughput)



$$NDCG_N = \frac{DCG_N}{IDCG_N}, \tag{20}$$

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

$$DCG_N = \sum_{j=1}^N \frac{2^{rel_j} - 1}{\log_2(j + 1)}, \tag{21}$$

where rel_j is the QoS value related to service ranked at j position. A high $NDCG_N$ expresses a high-accurate service ranking.

The next subsection analyzes the obtained results.

4.3 Results and analysis

For the experiments, the thresholds are set as $q_{c_0} = 2.5$, $q_{c_0+1} = 1.5$. At the process starting, the decay rate γ is set to 0.2; the number r of factors gradually increases from

5 to 50 in steps of 5. The matrix density is set to 5% and progressively raises to 60%.

In the next point, TASERM’s performances are compared to those of the other recommendation methods.

4.3.1 Performances analysis

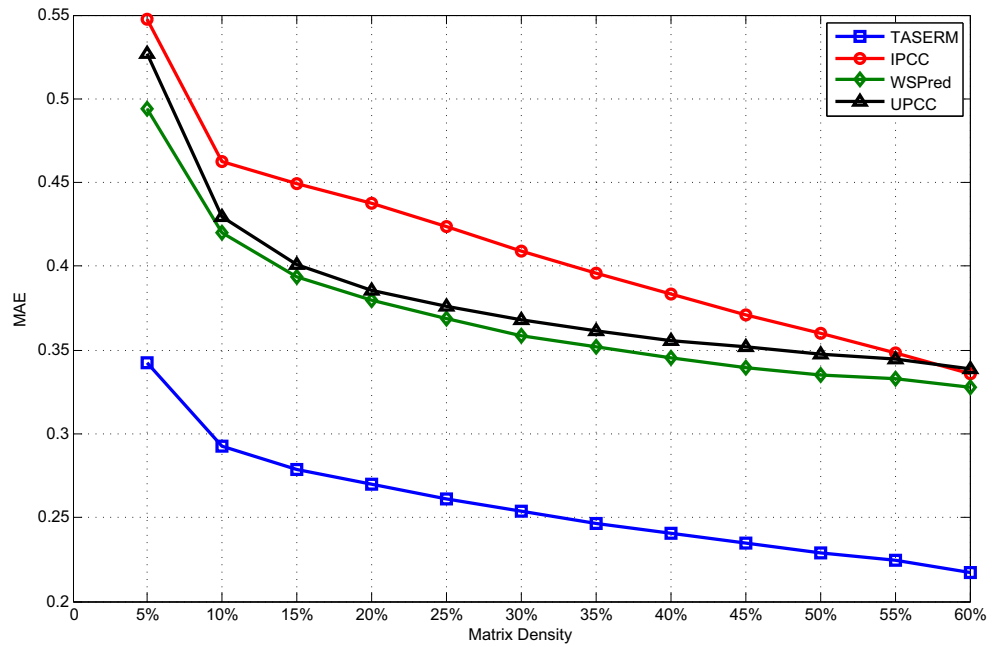
Our method is compared to the following recommendation methods:

- IPCC is a recommendation method that lays on item-based CF using Pearson Correlation Coefficient (PCC) for the inter-item similarities evaluation [30].
- UPCC is a recommendation method that lays on user-based CF using PCC for the inter-user similarities evaluation. Thereafter, the prediction is performed based on the weighted average of QoS values [30].
- WSPred is a time-aware recommendation method based on a personalized QoS prediction [28].

Table 1 Throughput performance evaluation at $r = 20$

Matrix Density (%)	50			55			60		
	MAE	RMSE	NDCG	MAE	RMSE	NDCG	MAE	RMSE	NDCG
TASERM	0.313	0.7391	0.9755	0.2698	0.6877	0.9785	0.2462	0.6606	0.9795
IPCC	1.1103	1.5723	0.8512	1.0894	1.5553	0.8551	1.0708	1.5417	0.8572
UPCC	0.8080	1.1649	0.8559	0.7943	1.15308	0.8594	0.78105	1.1424	0.8626
WSPred	0.6784	0.9994	0.9728	0.6479	0.9595	0.9773	0.6275	0.937	0.9793

Fig. 7 Matrix Density Impact on MAE Performances with $r = 20$ (response time)



Impact of the matrix density Following the throughput performances, Figs. 4, 5 and 6 illustrate the negative impact of data sparseness on the prediction precision. For each method, the QoS prediction accuracy is improved with the matrix density increase. Once more, the TASERM method significantly outperforms others methods even in the data sparseness case. Indeed, following Table 1 and Fig. 4, MAE performance of TASERM approach shows an improvement of 52.517% compared to IPCC, 35.092% compared to UPCC and 23.273 % compared to WSPred. Moreover, in

Fig. 5, RSME performance of TASERM approach displays an improvement of 43.928 % compared to IPCC, an increase of 24.307 % compared to UPCC and 13.111% compared to WSPred. In addition, in Fig. 6, NDCG performance of TASERM method is 9.533% better than IPCC, 8.879% better than UPCC, but decreases of 3.234 % compared to WSPred.

Following the response time performances, Figs. 7, 8 and 9 illustrate the negative influence of data sparsity on the QoS prediction accuracy and service ranking accuracy.

Fig. 8 Matrix Density Impact on RSME Performances with $r = 20$ (response time)

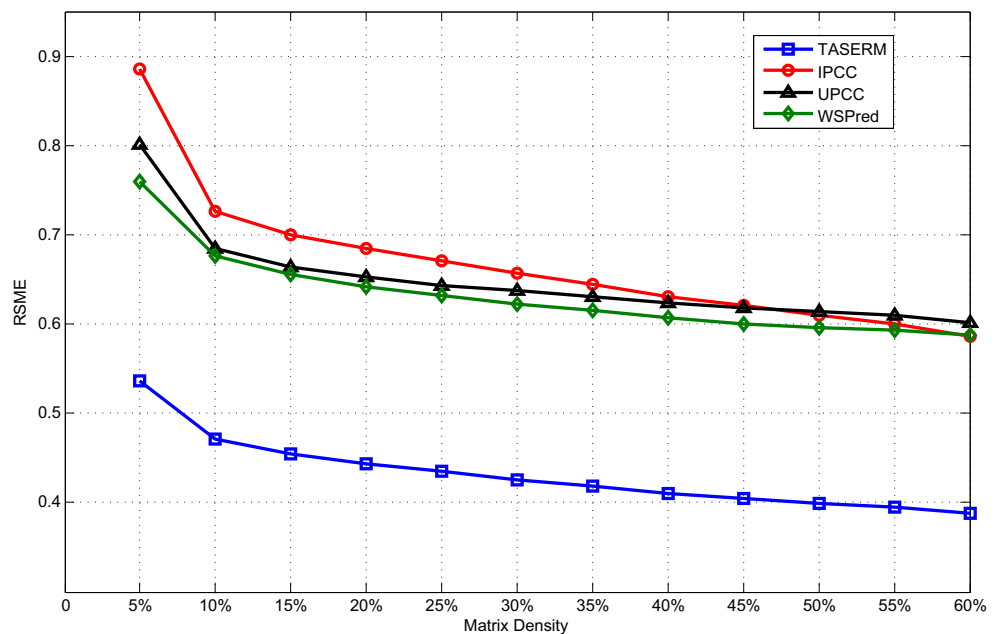
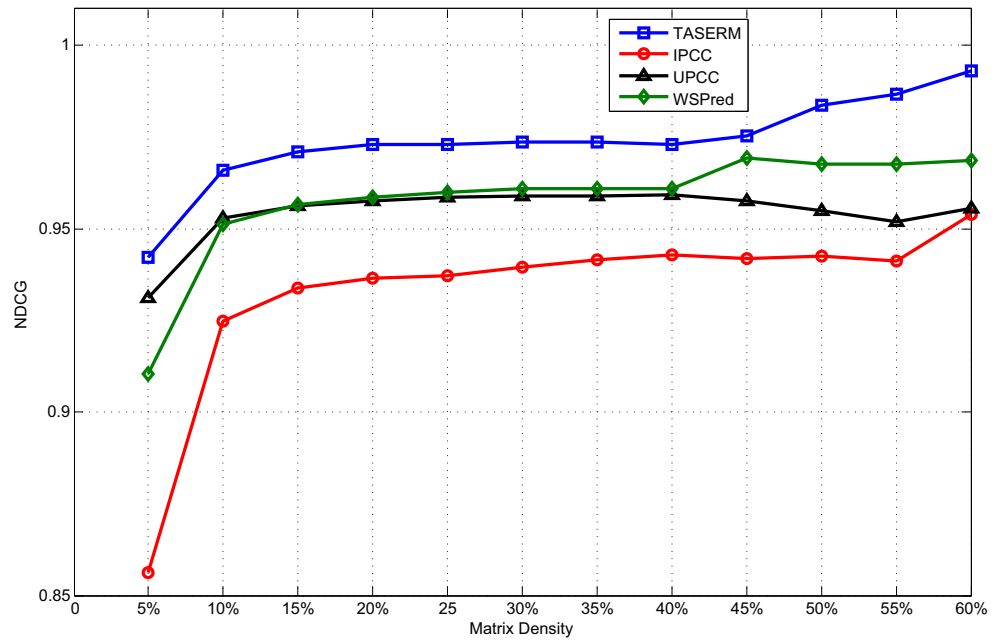


Fig. 9 Matrix Density Impact on NDCG Performances with $r = 20$ (response time)



Indeed, it can be observed that for each method, the QoS prediction is refined with the matrix density increase. However, the TASERM method significantly outperforms others methods even in the severe data sparseness situation. Indeed, following Table 2 and Fig. 7, MAE performance of TASERM approach shows an improvement of 37.21 % compared to IPCC, 32.61 % compared to UPCC and 30% compared to WSPred. Moreover, in Fig. 8, RSME performance of TASERM approach displays an improvement of 35.39 % compared to IPCC, 33.40 % compared to UPCC and 30% compared to WSPred. In addition, in Fig. 9, NDCG performance of TASERM method is 4.35% better than IPCC, 1.86% better than UPCC and 1.61% better than WSPred.

Impact of the number of factors To study the impact of the number of factors r , we configure r to 5; then it gradually increases to 50 in steps of 5. We propose to study TASERM performances at matrix densities equal to 15% and 60%.

Figures 10, 11 and 12 shows the impact of the number of factors. In these figures, it can be observed that TASERM's performances are maximized for $r = 20$. Indeed, in Figs. 10 and 11, for $r = 20$, MAE and RSME performance are minimal, meaning a high-accurate prediction while in Fig. 12, NDCG trend is maximal, meaning a high-accurate service ranking. This means that for a number of factors greater than 20, the additional data constitutes noise which affects the quality of prediction.

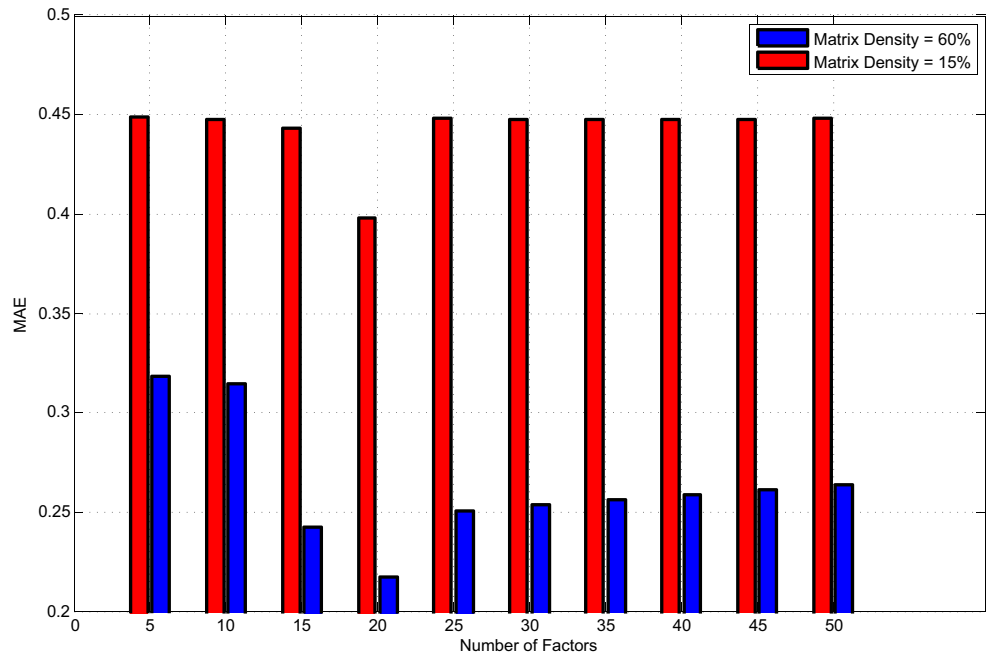
5 Conclusion and perspectives

In this paper, we proposed a data sparseness robust service recommendation approach which aims to guarantee in a sustainable way the relevance of services returned to end-users. Our method first performs the QoS prediction of the current time using a flexible factorization matrix technique applied to a set of instantaneous two-dimensional user-service matrices. Thereafter, the QoS prediction of

Table 2 Response time performance evaluation at $r = 20$

Matrix Density (%)	50			55			60		
	Algorithms	MAE	RMSE	NDCG	MAE	RMSE	NDCG	MAE	RMSE
TASERM	0.229194	0.399042	0.98381171	0.224784	0.394821	0.986705865	0.217539	0.387765	0.9930522
IPCC	0.35973	0.60975	0.942532	0.34857	0.60057	0.941219	0.33561	0.58554	0.9539
UPCC	0.34776	0.61362	0.955157	0.34461	0.6093	0.953339	0.33894	0.60174	0.9557
WSPred	0.32742	0.57006	0.9677076	0.32112	0.56403	0.96754551	0.31077	0.55395	0.96857279

Fig. 10 Factors number impact on MAE performances



the future time interval is performed using a time series forecasting method based on an AutoRegressive Integrated Moving Average (ARIMA) model. Based on past user-side QoS observations, the QoS prediction for the current time slot indicates the service relevance in the current time while the QoS prediction for the future time slot provides information on the anticipated service relevance relatively to the active user’s needs. The service recommendation is

based on the pair of criteria (q_{c_0}, q_{c_0+1}) which aims to ensure the sustainable appropriateness of services returned to the active user. Experiments conducted on a real-world dataset show the high-performances of our method compared to competing approaches. Since the rapid growth of big data offers increased possibilities and important sources of information to integrate into the recommendation process, we plan in the future, to exploit other data sources

Fig. 11 Factors number impact on RSME performances

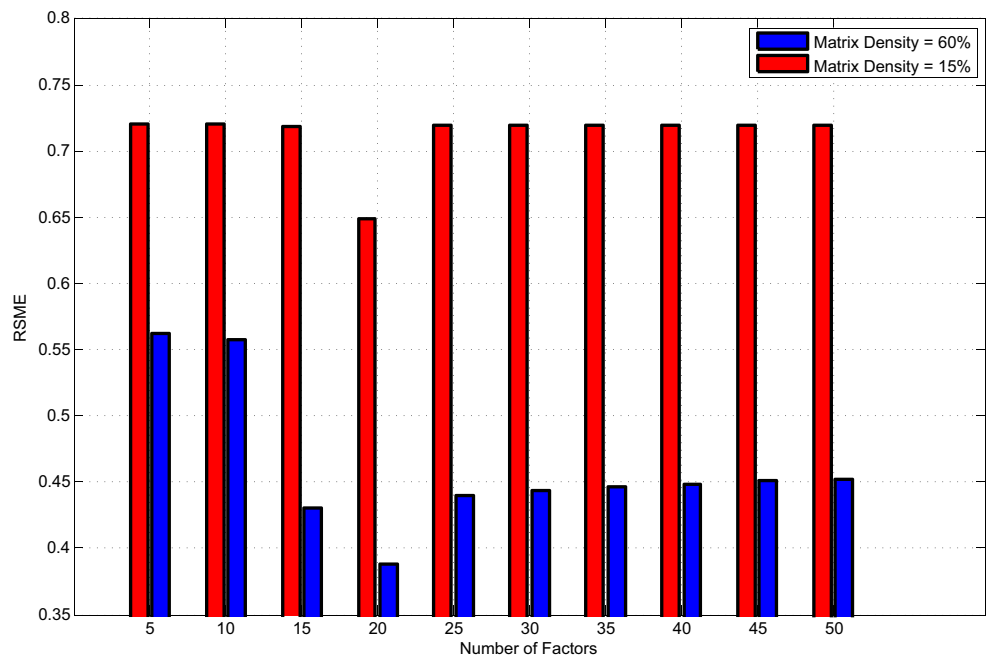
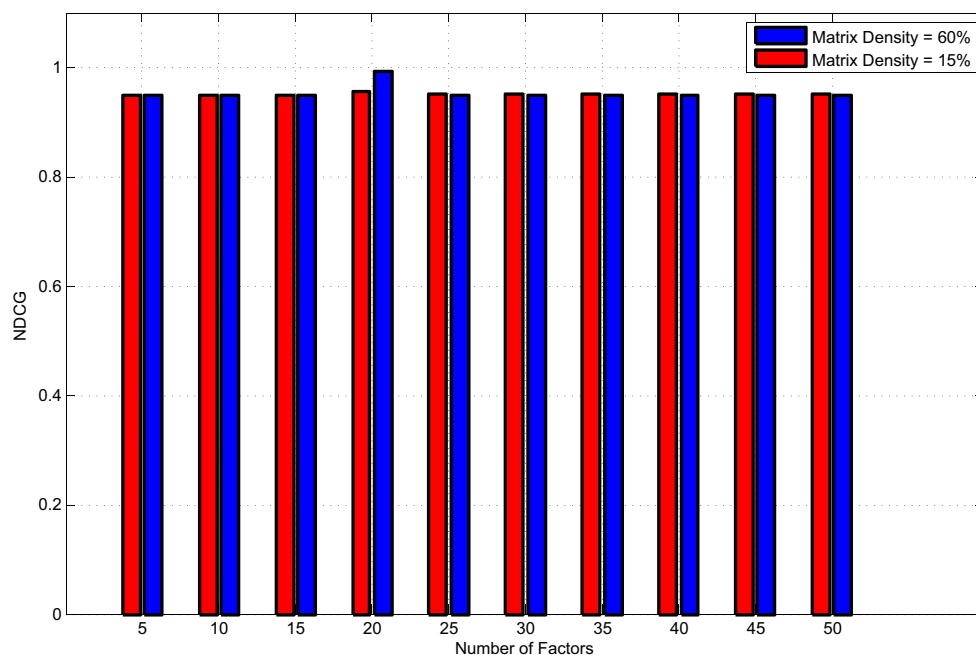


Fig. 12 Factors number impact on NDCG performances



to refine the service targeting and to get closer to end-users' expectations.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

- Ahn CK, Shmaliy YS, Zhao S (2018) A new unbiased fir filter with improved robustness based on frobenius norm with exponential weight. *IEEE Trans Circuits Syst II Express Briefs* 65(4):521–525. <https://doi.org/10.1109/TCSII.2017.2749006>
- Bokde D, Girase S, Mukhopadhyay D (2015) Matrix factorization model in collaborative filtering algorithms: A Survey. *Procedia Comput Sci* 49:136–146. <https://doi.org/10.1016/j.procs.2015.04.237>. <http://www.sciencedirect.com/science/article/pii/S1877050915007462>
- Box G, Jenkins GM (1976) *Time series analysis: Forecasting and control*. Holden-Day
- Campos PG, Díez F, Cantador I (2014) Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols. *User Model User-Adap Inter* 24(1):67–119. <https://doi.org/10.1007/s11257-012-9136-x>
- Ding S, Li Y, Wu D, Zhang Y, Yang S (2018) Time-aware cloud service recommendation using similarity-enhanced collaborative filtering and ARIMA model. *Dec Support Syst* 107:103–115. <https://doi.org/10.1016/j.dss.2017.12.012>. <http://www.sciencedirect.com/science/article/pii/S0167923617302415>
- Govindarajulu Z (1992) *Rank Correlation Methods* (5th edn). *Technometrics* 34(1):108–108. <https://doi.org/10.1080/00401706.1992.10485252>
- Hernandez-Matamoros A, Fujita H, Hayashi T, Perez-Meana H (2020) Forecasting of covid19 per regions using arima models and polynomial functions. *Appl Soft Comput* 96:106610
- Hernando A, Bobadilla J, Ortega F (2016) A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowl-Based Syst* 97:188–202. <https://doi.org/10.1016/j.knosys.2015.12.018>. <http://www.sciencedirect.com/science/article/pii/S0950705115005006>
- Hu Y, Peng Q, Hu X, Yang R (2015) Web service recommendation based on time series forecasting and collaborative filtering. In: 2015 IEEE international conference on web services, pp 233–240. <https://doi.org/10.1109/ICWS.2015.40>
- Jayapriya K, Mary NAB, Rajesh RS (2016) Cloud service recommendation based on a correlated QoS ranking prediction. *J Netw Syst Manag* 24(4):916–943. <https://doi.org/10.1007/s10922-015-9357-5>
- Kluser D, Ekstrand MD, Konstan JA (2018) Rating-based collaborative filtering: Algorithms and evaluation. In: Brusilovsky P, He D (eds) *Social Information Access: Systems and Technologies*. Springer International Publishing, Cham, pp 344–390. https://doi.org/10.1007/978-3-319-90092-6_10
- Koohi H, Kiani K (2016) User based Collaborative Filtering using fuzzy C-means. *Measurement* 91:134–139. <https://doi.org/10.1016/j.measurement.2016.05.058>. <http://www.sciencedirect.com/science/article/pii/S0263224116302159>
- Lei X, Tie J, Fujita H (2020) Relational completion based non-negative matrix factorization for predicting metabolite-disease associations. *Knowl-Based Syst* 204:106238
- Li D, Chen C, Lv Q, Shang L, Zhao Y, Lu T, Gu N (2016) An algorithm for efficient privacy-preserving item-based collaborative filtering. *Future Gener Comput Syst* 55:311–320. <https://doi.org/10.1016/j.future.2014.11.003>. <http://www.sciencedirect.com/science/article/pii/S0167739X14002374>
- Luo X, Xia Y, Zhu Q (2013) Applying the learning rate adaptation to the matrix factorization based collaborative filtering. *Knowl-Based Syst* 37:154–164. <https://doi.org/10.1016/j.knosys.2012.07.016>. <http://www.sciencedirect.com/science/article/pii/S0950705112002043>
- Valdiviezo-Díaz P, Ortega F, Cobos E, Lara-Cabrera R (2019) A collaborative filtering approach based on Naï Bayes classifier. *IEEE Access* 7:108581–108592. <https://doi.org/10.1109/ACCESS.2019.2933048>

17. Qi L, Wang R, Hu C, Li S, He Q, Xu X (2019) Time-aware distributed service recommendation with privacy-preservation. *Inform Sci* 480:354–364. <https://doi.org/10.1016/j.ins.2018.11.030>, <http://www.sciencedirect.com/science/article/pii/S0020025518309186>
18. Li S, Wen J, Luo F, Ranzi G (2018) Time-Aware QoS prediction for cloud service recommendation based on matrix factorization. *IEEE Access* 6:77716–77724. <https://doi.org/10.1109/ACCESS.2018.2883939>
19. Sánchez-Moreno D, Zheng Y, Moreno-garcía MN (2020) Time-aware music recommender systems: Modeling the evolution of implicit user preferences and user listening habits in a collaborative filtering approach. *Appl Sci* 10(15):5324
20. Stuart A (1956) Rank Correlation Methods. By M. G. Kendall, 2nd edn. *British J Stat Psychol* 9(1):68–68. <https://doi.org/10.1111/j.2044-8317.1956.tb00172.x>
21. Neammanee T, Maneeroj S (2018) Time-Aware recommendation based on user preference driven. In: 2018 IEEE 42nd annual computer software and applications conference (COMPSAC), vol 02, pp 26–31. <https://doi.org/10.1109/COMPSAC.2018.10198>
22. Chen X, Zheng Z, Liu X, Huang Z, Sun H (2013) Personalized qos-Aware Web service recommendation and visualization. *IEEE Trans Serv Comput* 6(1):35–47. <https://doi.org/10.1109/TSC.2011.35>
23. Wu X, Fan Y, Zhang J, Lin H, Zhang J (2019) QF-RNN: QF-Matrix factorization based RNN for time-aware service recommendation. In: 2019 IEEE international conference on services computing (SCC), pp 202–209. <https://doi.org/10.1109/SCC.2019.00042>
24. Yu X, Jiang F, Du J, Gong D (2017) A user-Based cross domain collaborative filtering algorithm based on a linear decomposition model. *IEEE Access* 5:27582–27589. <https://doi.org/10.1109/ACCESS.2017.2774442>
25. Zheng X, Xu LD, Chai S (2017) Qos recommendation in cloud services. *IEEE Access* 5:5171–5177. <https://doi.org/10.1109/ACCESS.2017.2695657>
26. Hu Y, Peng Q, Hu X (2014) A Time-aware and data sparsity tolerant approach for web service recommendation. In: 2014 IEEE international conference on web services, pp 33–40. <https://doi.org/10.1109/ICWS.2014.18>
27. Koren Y, Bell R, Volinsky C (2009) Matrix factorization techniques for recommender systems. *Computer* 42(8):30–37. <https://doi.org/10.1109/MC.2009.263>
28. Zhang Y, Zheng Z, Lyu MR (2011) WSPred: A time-aware personalized QoS prediction framework for web services. In: 2011 IEEE 22nd international symposium on software reliability engineering, pp 210–219. <https://doi.org/10.1109/ISSRE.2011.17>
29. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM (2013) Time-aware Point-of-interest Recommendation. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval, ACM, New York, NY, USA, SIGIR '13, pp 363–372. <https://doi.org/10.1145/2484028.2484030>, event-place: Dublin, Ireland
30. Yang Z, Wu B, Zheng K, Wang X, Lei L (2016) A survey of collaborative filtering-Based recommender systems for mobile internet applications. *IEEE Access* 4:3273–3287. <https://doi.org/10.1109/ACCESS.2016.2573314>
31. Zheng Z, Ma H, Lyu MR, King I (2011) Qos-aware web service recommendation by collaborative filtering. *IEEE Trans Serv Comput* 4(2):140–152. <https://doi.org/10.1109/TSC.2010.52>
32. Zheng Z, Wu X, Zhang Y, Lyu MR, Wang J (2013) Qos ranking prediction for cloud services. *IEEE Trans Parallel Distrib Syst* 24(6):1213–1222. <https://doi.org/10.1109/TPDS.2012.285>
33. Zhou K, Yang SH, Zha H (2011) Functional matrix factorizations for cold-start recommendation. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval, association for computing machinery, New York, NY, USA, SIGIR '11, pp 315–324. <https://doi.org/10.1145/2009916.2009961>, event-place: Beijing, China

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Armielle Noulapeu Ngaffo

obtained a Master's degree in telecommunication services engineering in 2016 at the National Advanced School of Posts, Telecommunications and ICT (SUP'PTIC) in Yaoundé, Cameroon. She is currently a PhD student at the Higher School of Communications of Tunis (SUP'COM) and is a member of Mediatron laboratory. Her research interests are recommender systems, sentiment analysis, artificial intelligence, services

discovery/exposure, big data analysis, cloud computing, virtualization, service-oriented architecture, services engineering.



Walid El Ayeb

is currently a teacher and researcher affiliated to Mediatron Laboratory. He received a PhD degree from the Higher School of Communications of Tunis (SUP'COM), University of Carthage in 2017. He received a degree in Software Engineering from INSAT (National Institute of Applied Sciences and Technology) in 2011. 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, Internet of Things, 4G mobile communication, graph theory, 5G mobile communication, and IP multimedia subsystem.



Zied Choukair is currently a Full Professor at Higher School of Communications of Tunis (SUP'COM), University of Carthage. He is affiliated to Mediatron Laboratory. He received a PhD degree at the University of Paris-Sud XI. He was an Associate Professor in computer science at ENST (National School of Telecommunications of Brittany) (in 2003). He was an Associate Professor in computer science and researcher at the IT Research Institute of Toulouse

(in 2005). 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, Internet of things, mobile communication systems, IP networks, fuzzy set theory, 4G mobile communication, 5G mobile communication, and IP multimedia subsystem.