Trace Norm Regularized Matrix Factorization for Service Recommendation

Qi Yu  
Rochester Institute of Technology  
qi.yu@rit.edu

Zibin Zheng  
The Chinese University of Hong Kong  
zbzheng@cse.cuhk.edu.hk

Hongbing Wang  
Southeast University, China  
hbw@seu.edu.cn

Abstract—We present in this paper a novel QoS prediction approach to tackle service recommendation, which is to recommend services with the best QoS to users. QoS prediction exploits available QoS information to estimate users’ QoS experience from previously unknown services. In this regard, it can be modeled as a general matrix completion problem, which is to recover a large QoS matrix from a small subset of QoS entries. The infinite number of ways to complete an arbitrary QoS matrix makes the problem extremely ill posed. The highly sparse QoS data further complicates the challenges. Nonetheless, real-world QoS data exhibits two key features, which can be leveraged for accurate QoS predictions, leading to high-quality service recommendations. First, QoS delivery can be significantly affected by a small number dominant factors in the service environment (e.g., communication link and user-service distance). Hence, it is natural to assume that the QoS matrix has a low-rank or approximately low-rank structure. Second, users (or services) that share common environmental factors are expected to receive (or deliver) similar QoS and hence can be grouped together. The proposed approach seamlessly amalgamates these two features into a unified objective function and employs an effective iterative algorithm to approach the optimal completion of an arbitrary QoS matrix. We conduct a set of experiments on real-world QoS data to demonstrate the effectiveness of the proposed algorithm.

Keywords—Collaborative filtering; matrix completion; trace norm regularization; matrix factorization

I. INTRODUCTION

Service-oriented computing offers an attractive paradigm for the provisioning and consuming of computing resources for a wide spectrum of domains. The large number of applications heavily taking advantage of this new computing paradigm has lead to the deployment of substantial Web services [1]. Many Web services may also offer similar functionalities but vary from each other in terms of the Quality of Service (QoS) they deliver [11], [16]. The QoS is mainly made of user centered quality parameters and examples include availability, response time, and so on. Meanwhile, Web services are autonomous entities as they are developed and deployed by independent developers in an open online environment. Therefore, some services may not deliver on what they promise.

Realizing the full potential of service computing requires the ability to efficiently and accurately retrieve services that meet users’ QoS requirements. However, the key challenge arises as different users may receive distinct QoS from the same provider given that users may come from different network environment, locate in different places, access the service in different time during the day, and so on. Since there are a large number of candidate services available online, casual service users do not have the time, resource, and expertise to evaluate the QoS of all available services and make an optimal choice. Therefore, a systematic approach is in demand to automatically estimate the QoS of services that users have not used before and recommend the most suitable ones to them.

Inspired by the recent success of recommendation systems in e-commerce (e.g., Amazon) and online content distribution (e.g., Netflix), similar techniques have been adopted in service computing for making service recommendations [10], [14], [19], [20]. As the backbone technology of many recommendation systems, Collaborative Filtering (CF) takes into consideration user disparity and leverages similar users’ QoS experience to predict the QoS of services that are previously unknown to an active user. Existing CF techniques can be generally divided into two categories: neighborhood-based and model based. The former requires to locate users who share similar QoS experience with the active user based upon the QoS from services commonly invoked by these users. As each user may invoke a limited number of services, it is usually difficult to identify a reasonable number of users that have invoked a common set of services. Hence, the neighborhood-based approaches suffer from the QoS data sparsity issue [15]. Model based approaches construct a global model based on the observed QoS data, which can be used to make QoS predictions. Nonetheless, the limited observable QoS data poses key challenges for building accurate prediction models. The high computational overhead caused by the tuning of a large number of model parameters usually limits the scalability of many model based approaches.

In this paper, we develop a novel collaborative filtering algorithm for making QoS-aware service recommendations. The proposed algorithm effectively tackles the QoS data sparsity issue, which leads to accurate QoS predictions. Specifically, by indexing users with rows and services with columns, we form a matrix $M$ of QoS data where $M_{ij}$ denote the QoS that user $u_i$ received from service $s_j$. Since each user may just invoke a limited number of services, only a subset of QoS entries, $\{M_{ij} : (i,j) \in \Omega\}$, are
available, where \( \Omega \) is the set of indices for the observed QoS entries. Hence, we model QoS prediction as a matrix completion problem, which is to recover a large QoS matrix from a small subset of observed QoS entries. However, the matrix completion problem is extremely ill posed as there are infinite ways to complete the QoS matrix \( M \). Hence, accurate completion of a QoS matrix with many missing entries poses non-trivial challenges.

The proposed collaborative filtering algorithm achieves accurate QoS prediction by leveraging two inherent features of real-world QoS data:

- A small number of latent factors in the online environment may significantly affect the QoS delivery from services to users. Such factors may include the communication links, physical distances between users and services, hardware/software configurations from both user and service sides, and so on. Hence, it is natural to assume that the QoS matrix has a low-rank or approximately low-rank structure. Specifically, given a partially observed QoS matrix \( M \), we can recover a complete matrix \( X \) by minimizing its rank, i.e., \( \min_X \text{rank}(X) \), while ensuring that \( X \) and \( M \) agree on the observed entries, i.e., \( X_{ij} = M_{ij} \), where \( (i, j) \in \Omega \).
- Users (or services) that share common online environmental factors are expected to receive (or deliver) similar QoS and hence can be grouped together. Hence, the QoS matrix can be expressed using a clustered representation. In particular, the user and service groups can be achieved by performing Non-negative Matrix Tri-Factorization (NMTF) on a QoS matrix \( X \), which clusters users and services, simultaneously. Therefore, the QoS matrix \( X \) is expected to have a NMTF representation.

A complete QoS matrix can be recovered by solving a unified objective function that captures the above two features. Nonetheless, directly solving the unified objective function is NP-hard as the first feature leads to a rank minimization problem, which is NP-hard by itself. Fortunately, the rank function of a matrix can be well approximated by its trace norm, which is the sum of the singular values of the matrix [8]. Since the trace norm is a convex function of a matrix, it provides an effective heuristic to solve the non-convex rank minimization problem.

We integrate the trace norm as a regularization component into the NMTF process and develop the Trace Norm Regularized Matrix Factorization (TNR-MF) algorithm for QoS prediction. By minimizing a trace norm regularized least square function, TNR-MF achieves a complete QoS matrix that best matches a given NMTF representation. However, the central challenge is that the NMTF representation of a complete QoS matrix is unknown in advance. TNR-MR exploits an iterative process that performs matrix completion and NMTF representation computation in turn, which discovers the complete QoS matrix along with its NMTF representation at convergence. Experiments on real-world QoS data demonstrate that TNR-MF outperforms existing competitive service recommendation algorithms, including both neighborhood-based and model based ones.

The remainder of the paper is organized as follows. We review some existing works that are most relevant to ours in Section II. We describe in detail the proposed TNR-MF algorithm in Section III. We assess the effectiveness of the proposed algorithm via a set of experiments on real QoS data in Section IV and conclude in Section V.

II. RELATED WORK

The exponential growth of the number of Web services poses key challenges to discover services that satisfy user desired functional and non-functional requirements. Existing techniques that can be applied to service discovery fall into two broad categories: functionality-based and QoS-based. Functionality-based service discovery aims to locate services that satisfy user required functionality. Information retrieval techniques [7] and semantic technologies (e.g., using ontologies) [12] have been exploited to improve the accuracy of service discovery. QoS-based approaches, on the other hand, are used to differentiate service providers based on their QoS performance [16], [17].

Collaborative Filtering (CF) based techniques have been recently adopted to provide QoS-aware recommendations to service users [10], [14], [19], [20]. A standard user-based CF algorithm was developed in [14], which makes QoS prediction by assuming that similar users tend to receive similar QoS from similar services. A hybrid algorithm was proposed that enhances the user-based approach by integrating item-based CF to achieve better QoS prediction accuracy [20]. A number of similar algorithms have thereafter been developed that leverage other information, such as users’ locations [4], [5], invocation frequencies of services [13], and query histories of users [19] to improve the quality of service recommendation. Both user and item based approaches follow the neighborhood centric strategy in CF, which explores the local neighborhood to identify similar users or for recommendation. A model based CF algorithm is recently developed that achieves higher prediction accuracy [21]. The proposed algorithm uses the user-based approach as a precursor to identify top-k similar users. Based on the user neighborhood information, matrix factorization is employed to construct a global model, which can be used to predict unobserved QoS data.

As discussed in Section II, QoS prediction can be modeled as a general matrix completion problem, which finds applications in many science and engineering disciplines. A commonly used strategy to solve such an ill-posed problem is to assume that the matrix has a low-rank or approximately low-rank structure. However, rank minimization is NP-hard due to the non-convexity and discontinuity of the
rank function \(3, 13\). It has been demonstrated that the trace norm provides the tightest convex relaxation of the rank minimization problem \(8\). A number of optimization algorithms have been recently developed that apply trace norm as a convex surrogate to effectively tackle the matrix completion problem \(3, 9, 18\).

The proposed TNR-MF algorithm seeks a complete QoS matrix \(X\) that is not only low-rank but also captures the group structure of users and services (through NMTF). Therefore, it goes beyond just completing a matrix from a small subset of its entries as focused by existing approaches. Having a low-rank structure and a clustered representation is in line with the inherent features of a QoS matrix. Hence, an accurate matrix completion is expected to be achieved by TNR-MF. This is further justified through our extensive experiments on real-world QoS datasets. In particular, the better QoS prediction performance over one of the best matrix completion algorithms clearly demonstrates the usefulness of integrating clustered representation of QoS data, which also differentiates TNR-MF from all existing matrix completion algorithms.

III. THE TNR-MF ALGORITHM

We present the TNR-MF algorithm for QoS prediction in this section. TNR-MF aims to discover a complete QoS matrix from a small subset of observed QoS entries obtained from historical user-service interactions. TNR-MR leverages the low-rank structure and the clustered representation of QoS data and exploits an effective iterative process to achieve QoS matrix completion with high accuracy. In what follows, we start by introducing the rank minimization techniques for QoS matrix completion. We then describe how to integrate matrix factorization for group structure discovery, which leads to a unified objective function. An iterative algorithm is presented in the end for solving the unified objective function.

A. QoS Matrix Completion via Trace Norm Minimization

Before presenting the technical details, we first describe the symbols and notations that are used throughout the paper. Assume that there are \(m\) existing users and \(n\) Web services. The QoS attribute (e.g., response time, reliability, and availability) under consideration takes positive real values. We use a matrix \(M \in \mathbb{R}^{m \times n}\) to denote the observed QoS data, where \(M_{ij}\) represents the QoS that user \(u_i\) received from service \(s_j\). In this regard, the \(i\)-th row of \(M\) represents user \(u_i\) while the \(j\)-th column of \(M\) represents service \(s_j\). Let \(\Omega\) denote the set of indices for the observed QoS entries. The goal is to find a complete matrix \(X \in \mathbb{R}^{m \times n}_+\) that best estimates the missing QoS entries.

Since the QoS delivery from services to users is usually affected by a small number dominant factors in the online environment (e.g., communication link and user-service distance), it is natural to assume that the QoS matrix has a low-rank or approximately low-rank structure. Hence, the complete QoS matrix \(X\) can be computed by solving the following rank minimization problem:

\[
\begin{align*}
\text{minimize} & \quad \text{rank}(X) \\
\text{subject to} & \quad X_{ij} = M_{ij}, \forall (i, j) \in \Omega
\end{align*}
\]

(1) aims to recover a matrix \(X\) that captures the most influential latent factors affecting QoS delivery through rank minimization. The constraints make sure that the recovered low-rank matrix respects the observed QoS entries. Nonetheless, the optimization problem \(1\) is NP-hard due to the non-convexity and discontinuity of the rank function \(3, 18\). It has been demonstrated that the trace norm, which is defined as the sum of the singular values of the matrix, provides the tightest convex relaxation of the rank minimization problem \(8\). Hence, we can instead solve a trace norm minimization problem, which leads to a QoS matrix that provides a good approximation of the target low-rank matrix. Specifically, the trace norm minimization problem is formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad ||X||_* \\
\text{subject to} & \quad X_{ij} = M_{ij}, \forall (i, j) \in \Omega
\end{align*}
\]

(2) where \(||X||_* = \sum_{i=1}^{\min(n,m)} \sigma_i(X)\) and \(\sigma_i(X)\) is the \(i\)-th largest singular value of \(X\). The classical method for solving \(2\) is Semi-Definite Programming (SDP), which however can only deal with small matrices. Some efficient first-order algorithms, such as the Singular Value Thresholding (SVT) algorithm \(13\), have been recently developed that tackle the scalability issue and hence can be used to solve large-scale matrix completion problems.

B. Trace Norm and Matrix Factorization Integration

The trace norm minimization approach only considers the low-rank property of the QoS matrix when performing matrix completion. As discussed in Section II real-world QoS data also exhibits a group structure of users and services. Specifically, users and services that share common latent environmental factors are expected to receive and deliver similar QoS and hence can be grouped together, respectively. Since matrix data can be clustered through factorization of the matrix \(M\), we expect the QoS matrix to possess a factorized representation that explicitly shows the group structure of users and services. In this section, we present trace norm and matrix factorization integration, which leads to a unified objective function for QoS matrix completion.

Since both user and service groups are aimed to be discovered from the QoS data, we propose to use Non-negative Matrix Tri-Factorization (NMTF) to factorize the QoS matrix \(X\), which results in three matrices, \(U, S\) and \(R\), i.e., \(X = USR^T\). More specifically, \(U \in \mathbb{R}^{n \times k}\) is the user cluster indicator matrix, \(S \in \mathbb{R}^{m \times l}\) is the service cluster
indicator matrix (S' is the transpose of S), and R ∈ R^{k×l} is the cluster association matrix that captures the relationship between user clusters and service clusters. In this way, NMTF simultaneously clusters the m users into k disjoint user groups and n services into l disjoint service groups [6]. In what follows, we provide an intuitive interpretation of the NMTF based clustering, which helps understand why the user and service group structure is captured by the resultant factorized representation of X.

As users and services in the same clusters share similar latent environmental factors, they also receive or deliver similar QoS, respectively. In this regard, entry R_{qp} ∈ R essentially represents the overall QoS perception of the p-th cluster of users on the q-th cluster of services. Based on the factorization result, user u_i’s QoS perception on service s_j is given by

\[ X_{ij} = \sum_{p=1}^{k} U_{ip} \left( \sum_{q=1}^{l} R_{pq} S_{jq} \right) \]  

Assume that entry U_{ip} ∈ U denotes the cluster membership of user u_i in the p-th user cluster and entry S_{jq} ∈ S denotes the cluster membership of service s_j in the q-th service cluster. Hence, u_i’s QoS perception on s_j is exactly a linear combination of all user groups’ QoS perception on s_j, weighted by the cluster membership of u_i in these user groups, where \( \sum_{q=1}^{l} R_{pq} S_{jq} \) denotes the QoS perception of the p-th user group on s_j. To ensure that the total probability of assigning a user (or service) to different user (or service) clusters sums up to 1, we enforce the following two constraints on the user and service cluster memberships, respectively:

\[ \sum_{p=1}^{k} U_{ip} = 1, \forall i \in [1,n], \sum_{q=1}^{l} S_{jq} = 1, \forall j \in [1,m] \]  

Furthermore, entries in U and S should not take negative values as they denote membership probabilities. Since entities in R denote the overall group QoS perceptions, they should also take non-negative values.

By integrating the low-rank property with the clustered representation of the QoS data, we propose a unified objective function for solving QoS matrix completion:

\[ \begin{align*} 
\text{minimize} & \quad \frac{1}{2} \| X - U R S' \|_F^2 + \tau \| X \|_* \\
\text{subject to} & \quad X_{ij} = M_{ij}, \forall (i,j) \in \Omega \\
 & \quad U1 = 1, S1 = 1 \\
 & \quad U, R, S \succeq 0 
\end{align*} \]  

where \( \| X \|_F = \sqrt{\sum_{i,j} X_{ij}^2} \) is Frobenius norm of X and \( \tau \) is the regularization parameter that balances between the low-rank structure and the clustered representation of X. U1 = 1 and S1 = 1 are the matrix representation of (4).

C. The Optimization Algorithm

The clustered representation of matrix X (i.e., URS') remains unknown before X is discovered. Hence, existing trace norm minimization algorithms, such as SDP and SVT, are not applicable to (5), which requires to solve X and (U, R, S) simultaneously. We develop a novel algorithm that iteratively optimizes (5) by alternatively optimizing with respect to X (low-rank structure) and (U, R, S) (clustered representation) while holding the other fixed. For learning X, the optimization problem is a trace norm regularized least squares problem with linear constraints. This can be efficiently achieved by applying a simple gradient descent algorithm to the Lagrange dual of the problem due to the special property of the trace norm (see below for details). For learning (U, R, S), we derive efficient update rules, which alternatively update U, R, and S while keeping the other two fixed. The objective function monotonically decreases under these update rules, which ensures convergence to the optimal (U, R, S).

C.1 Computing Matrix Completion

Consider solving the optimization problem (5) with a linear constraint over X while keeping (U, R, S) fixed. Let C denote URS' and P_Ω be the orthogonal projection onto the span of matrices vanishing outside of Ω so that the \( P_Ω(X)_{ij} = X_{ij} \) if \((i,j) \in \Omega \) and \( P_Ω(X)_{ij} = 0 \) otherwise. Hence, (5) is reformulated as

\[ \begin{align*} 
\text{minimize} & \quad \frac{1}{2} \| X - C \|_F^2 + \tau \| X \|_* \\
\text{subject to} & \quad P_Ω(X) = P_Ω(M) 
\end{align*} \]  

The Lagrangian of the optimization problem (6) is given by

\[ \mathcal{L}(X, Y) = \frac{1}{2} \| X - C \|_F^2 + \tau \| X \|_* + \langle Y, P_Ω(X - M) \rangle \]  

where \( \langle X, Y \rangle = \text{trace}(X^T Y) \) is the inner product of two matrices and \( Y \in \mathbb{R}^{n \times n} \) is the Lagrangian variable. From (7), we derive the dual function as

\[ f(Y) = \inf_X \mathcal{L}(X, Y) \]  

Assume that \( Y^* \) is the optimal dual variable obtained by maximizing the dual function \( f(Y) \). Since strong duality holds for \( \mathcal{L}(X, Y) \) [5], we have

\[ \mathcal{L}(X^*, Y^*) = \inf_X \mathcal{L}(X, Y^*) \]  

\[ X^* = \arg\min_X \mathcal{L}(X, Y^*) \]  

where \( X^* \) is the optimal primal variable, which is the complete QoS matrix we aim to recover. Therefore, an optimal QoS completion can be achieved by first optimizing the dual function \( f(Y) \) in (8) to get the optimal dual variable \( Y^* \) and then computing \( X^* \) based on (9).
The dual function $f(Y)$ is a continuously differentiable concave function \[^2\]. Hence, we apply the following gradient step to approach $Y^*$ iteratively:

$$
Y_k = Y_{k-1} + \frac{1}{\lambda_{k-1}} \nabla f(Y_{k-1})
$$

(10)

where $\frac{1}{\lambda_{k-1}}$ is a positive step size. $\nabla f(Y_{k-1})$ is the gradient of $f(Y)$ evaluated at $Y_{k-1}$, which is given by

$$
\nabla f(Y_{k-1}) = \frac{\partial L(X_k, Y)}{\partial Y} = \mathcal{P}_\Omega(X_k - M)
$$

(11)

where $X_k$ is the minimizer of the Lagrangian for $Y_{k-1}$:

$$
X_k = \text{argmin}_X L(X, Y_{k-1})
$$

$$
= \text{argmin}_X \frac{1}{2} ||X - C - \mathcal{P}_\Omega(Y_{k-1})||_F^2 + \tau ||X||_*,
$$

$$
= \text{argmin}_X \frac{1}{2} ||X - \hat{C}||_F^2 + \tau ||X||_*
$$

(12)

where $\hat{C} = C + \mathcal{P}_\Omega(Y_{k-1})$. By exploring the special structure of the trace norm, $X_k$ can be obtained by computing the singular value decomposition (SVD) of $C$ and then applying soft-thresholding to the singular values of $\hat{C}$. Specifically, given a matrix $C \in \mathbb{R}^{m \times n}$, the SVD of $C$ is given by

$$
C = U \Sigma V^T, \quad \Sigma = \text{diag} \{ \{ \sigma_i \} \}
$$

(13)

where $1 \leq i \leq \min(m, n)$ and $\sigma_i$ is the $i$-th largest singular value of $C$. The singular value thresholding operator is defined as \[^3\]:

$$
\mathcal{D}_\tau(C) = UD_\tau(\Sigma)V', \quad D_\tau(\Sigma) = \text{diag} \{ \{ (\sigma_i - \tau)_+ \} \}
$$

(14)

where $(\sigma_i - \tau)_+ = \max((\sigma_i - \tau), 0)$. It has been proved in \[^3\] that (12) can be solved by applying the singular value thresholding operator to $\hat{C}$:

$$
X_k = \mathcal{D}_\tau(\hat{C})
$$

$$
= \text{argmin}_X \frac{1}{2} ||X - \hat{C}||_F^2 + \tau ||X||_*
$$

(15)

Intuitively, $\mathcal{D}_\tau(\hat{C})$ uses $\tau$ as a threshold value to eliminate singular values of $\hat{C}$ that are no larger than $\tau$. The minimizer $X_k$ will be a low-rank matrix due to the soft-thresholding applied to the singular values of $\hat{C}$.

Once $Y^*$ is achieved by iteratively applying the gradient step as described above, $X^*$ can be computed by solving (13). This is achieved by applying the singular value thresholding operator to $C^* = \hat{C} + \mathcal{P}_\Omega(Y^*)$, which leads to $X^* = \mathcal{D}_\tau(C^*)$.

### C.2 Computing Clustered Representation of QoS Data

In this subsection, we present a method for solving optimization problem (5) over $(U, R, S)$ given fixed complete QoS matrix $X$. We further convert constraints $U1 = 1$ and $S1 = 1$ into penalty terms, $\alpha||U1 - 1||^2$ and $\beta||S1 - 1||^2$, where $\alpha, \beta \geq 0$ are penalty parameters. This reduces to the following problem:

$$
\text{minimize} \frac{1}{2} ||X - URS'||_F^2 + \alpha||U1 - 1||^2 + \beta||S1 - 1||^2
$$

subject to $U, R, S \geq 0$

(16)

(16) is convex in $U, R,$ and $S$, respectively when the other two are fixed. Hence, we develop a set of update rules that alternatively update $U, R,$ and $S$ while keeping the other two matrices fixed. The clustered representation $(U, R, S)$ of a given QoS matrix $X$ will be obtained at the convergence of this iterative process. In what follows, we derive the update rules for $U, R,$ and $S$, respectively.

**Update Rule for $U$:** The update rule for $U$ aims to find an optimal $U$ given fixed $R$ and $S$. Extracting the $U$ relevant terms in (16) leads to the following problem:

$$
\text{minimize} \frac{1}{2} ||X - URS'||_F^2 + \alpha||U1 - 1||^2
$$

subject to $U \geq 0$

(17)

The Lagrangian of (17) is given by

$$
L(U, \Phi) = \frac{1}{2} ||X - URS'||_F^2 + \alpha||U1 - 1||^2 + \langle \Phi, U \rangle
$$

(18)

where $\Phi \in \mathbb{R}^{m \times k}$ is the Lagrangian variable. The partial derivative of $L(U, \Phi)$ respect to $U$ is as expressed follows:

$$
\frac{\partial L(U, \Phi)}{\partial U} = -2XSR' + 2URS'SR' + 2\hat{\alpha}(UE - E) + \Phi
$$

(19)

where $E \in \mathbb{R}^{k \times k}$ is a matrix of all 1’s and $\hat{\alpha} = \alpha/k$. We use the fact $\alpha||U1 - 1||^2 = \hat{\alpha}||UE - E||^2$ for the above derivation. At convergence, (19) is set to 0. Hence,

$$
\Phi = 2XSR' - 2URS'SR' - 2\hat{\alpha}(UE - E)
$$

(20)

Also, at convergence, the KKT complementarity condition for the non-negativity $\Phi \odot U = 0$ holds, where $\odot$ is Hadamard product (i.e., element-wise product) of two matrices. This leads to

$$
(2XSR' - 2URS'SR' - 2\hat{\alpha}(UE - E)) \odot U = 0
$$

(21)

From (21), we derive the update rule for $U$ as follows

$$
U_{ip} = U_{ip} \left( \frac{\langle XSR' \rangle_{ip} + \hat{\alpha}}{\langle \hat{\alpha}UE + URSSR' \rangle_{ip}} \right)^{1/2}
$$

(22)

**Update Rule for $S$:** The update rule for $S$ can be derived by solving the following problem:

$$
\text{minimize} \frac{1}{2} ||X - URS'||_F^2 + \beta||S1 - 1||^2
$$

subject to $S \geq 0$

(23)
Applying the Lagrangian multiplier method similar to the one described above solves (24), which leads to the update rule for $S$:

$$S_{jq} = S_{jq} \left( \frac{(X'UR)_{jq} + \beta}{\beta SE + SR'UR} \right) ^{\frac{1}{2}}$$

(24)

where $E \in \mathbb{R}^{l \times l}$ is a matrix of all 1’s and $\hat{\beta} = \beta/l$.

**Update Rule for $R$:** Along the same lines, we apply the Lagrangian multiplier method to the following problem:

$$\text{minimize} \quad \frac{1}{2} \| X - URS' \|^2_F$$

subject to $R \geq 0$

(25)

and derive the update rule for $R$ as follows:

$$R_{pq} = R_{pq} \left( \frac{(U'XS)_{pq}}{(U'URS'S)_{pq}} \right) ^{\frac{1}{2}}$$

(26)

IV. Experiments

We conduct a set of experiments to assess the effectiveness of the proposed TNR-MT algorithm for QoS prediction. The experiments are conducted on two real-world QoS datasets [20], [22]: QoS_1 and QoS_2. Both QoS datasets are collected by exploiting the large-scale compute nodes from the Planet-Lab [1], which currently consists of 1166 nodes at 549 sites. These compute nodes are leveraged to automatically invoke over several thousands Web services that are publicly available across the world. The two QoS datasets are further detailed as follows:

- **QoS_1:** 150 compute nodes from the Planet-Lab are used to invoke a hundred selected Web services. Each compute node invokes each service for 100 times and 1.5 million service invocation records are collected. The averaged Round-Trip Time (RTT) is used in our experiments.

- **QoS_2:** This dataset has a much larger scale, in which 339 compute nodes from the Planet-Lab are used to invoke 5,825 Web services. Each compute node invokes each service for one time and the obtained RTT is used in our experiments.

A. Experiment Design

For QoS_1, we organize the data entries into a $150 \times 100$ matrix $A$, in which entry $A_{ij}$ denotes the averaged RTT that user $i$ used to invoke service $j$. We randomly remove a certain percentage (75%–95%) of entries from $A$, resulting in a matrix $M$ to simulate a real-world QoS dataset, where only a small subset of entries are observed. We then compare the removed RTT entries with the predicted values to evaluate the accuracy of QoS prediction. For QoS_2, we organize the data into a $339 \times 5828$ matrix and then apply the same strategy to remove entries and assess the result of QoS prediction.

We employ Mean Absolute Error (MAE), one of the most widely used metric in recommendation systems, to assess the accuracy of QoS prediction:

$$MAE = \frac{1}{N} \sum_{i,j} \frac{|A_{ij} - \hat{X}_{ij}|}{N}$$

(27)

where $A_{ij}$ and $X_{ij}$ denote the actual and estimated RTT, respectively. $N$ is the total number of estimated QoS entries. The default numbers of user and service groups are 30 for QoS_1 and 80 for QoS_2. The default value for the regularization parameter $\tau$ is set to 1 for both datasets. These default values will be used in all the experiments unless specified otherwise. Since the proposed TNR-MT algorithm and some other algorithms under comparison in the experiments involves clustering of the QoS data, we perform a column normalization on the QoS matrix before applying the algorithms to avoid that the clustering result is dominated by columns with very large values.

B. QoS Prediction Performance

To demonstrate the effectiveness of the proposed TNR-MF algorithm, we implemented six competitive collaborative filtering methods for QoS prediction, which include both neighborhood-based and model-based approaches. These algorithms are detailed as follows:

- **UPCC** is a user based algorithm using Pearson Correlation Coefficient as similarity measure [14].

- **UCOS** is a user based algorithm using Cosine distance as similarity measure.

- **IPCC** is an item based algorithm using Pearson Correlation Coefficient as similarity measure.

- **WSRec** is a hybrid collaborative algorithm that combines both user and item based approaches using their prediction accuracy as the aggregation weights [20].

- **WNMTF** is a NMTF based approach that co-clusters users and items and uses the cluster structure for QoS prediction [15]. A weighting mechanism is adopted that minimizes a new objective function $\| W \odot (M - URS') \|^2_F$, where $W_{ij} = 1$ if $(i,j) \in \Omega$ and 0 otherwise.

- **SVT** is the singular value thresholding algorithm that achieves matrix completion via trace norm minimization [3].

Table I compares the MAE performance from the proposed TNR-MF algorithm with those from the six representative collaborative filtering algorithms. We reach several important observations. First, the MAE performance of TNR-MF outperforms those from WNMTF and SVT. This clearly justifies the effectiveness of integrating the low-rank structure with a clustered representation of the QoS data.
Second, for less sparse QoS data (e.g., when the missing rate is no more than 90%), the MAE performance of TNR-MR is comparable with the best neighborhood-based method, WSRec, which integrates the user-based and item-based algorithms and leverages the advantages of both. However, as QoS data becomes more sparse, the MAE performances of all neighborhood-based approaches decrease significantly. This confirms that neighborhood-based approaches suffer from data sparsity, making them less attractive for service recommendation given the extremely scarce observable QoS data. The model-based approaches, especially those that leverage clustering-based models, including WMNTF and TNR-MF, are much more robust to data sparsity. Given a cluster structure, two users can be considered as similar by receiving similar QoS from services in the same service cluster instead of having to receive similar QoS from the same service. In this way, the user and service clusters essentially serve as a bridge to relate users and services and hence effectively address data sparsity.

To demonstrate how the proposed algorithm perform on large-scale QoS prediction, Table I reports the MAE performance on QoS_2. We exclude all the neighborhood-based algorithms in this dataset because such algorithms need to perform similarity search for each individual user based upon each service, whose QoS needs to be predicted. For large-scale QoS prediction, these algorithms run extremely slow. In contrast, the model-based approaches, such as SVT and WMNTF, exploit simple gradient steps or efficient update rules, to compute all missing QoS entries simultaneously, which exhibit excellent computational scalability. Also the results from QoS_1 further confirms that neighborhood-based algorithms suffer from the data sparsity issue. Hence, we focus on comparing model-based algorithms.

We change the numbers of user and service clusters to 80 to accommodate the much larger number of users and services and keep all other parameters the same. It is clear from Table II that TNR-MF achieves the best MAE performance over different missing rates. The result also confirms that these model-based approaches are robust to data sparsity as the MAE performances of TNR-MF and WMNTF only drop slightly as the missing rate increases. SVT even achieves better MAE for some more sparse QoS data. This may be because the rank of the completed matrix discovered from the sample QoS entries matches better with the rank of the original QoS matrix in these test cases.

### Table I

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>MAE Performance on QoS_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing rate</td>
<td>75% 80% 85% 90% 95%</td>
</tr>
<tr>
<td>UPCC</td>
<td>0.0408 0.0443 0.0518 0.0647 0.0983</td>
</tr>
<tr>
<td>UCOS</td>
<td>0.0398 0.0438 0.0472 0.0604 0.0804</td>
</tr>
<tr>
<td>IPCC</td>
<td>0.0359 0.0367 0.0415 0.0455 0.0739</td>
</tr>
<tr>
<td>WSRRec</td>
<td>0.0350 0.0359 0.0405 0.0440 0.0713</td>
</tr>
<tr>
<td>WMNTF</td>
<td>0.0350 0.0383 0.0462 0.0522 0.0896</td>
</tr>
<tr>
<td>SVT</td>
<td>0.0607 0.0605 0.0696 0.0769 0.0928</td>
</tr>
<tr>
<td>TNR-MF</td>
<td>0.0350 0.0338 0.0441 0.0447 0.0588</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>MAE Performance on QoS_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missing rate</td>
<td>75% 80% 85% 90% 95%</td>
</tr>
<tr>
<td>WMNTF</td>
<td>0.0404 0.0480 0.0504 0.0513 0.0543</td>
</tr>
<tr>
<td>SVT</td>
<td>0.0804 0.0742 0.0769 0.0657 0.0697</td>
</tr>
<tr>
<td>TNR-MF</td>
<td>0.0486 0.0475 0.0499 0.0501 0.0537</td>
</tr>
</tbody>
</table>

### C. Impact of Parameters

We investigate the impact of two important parameters in this section, including the numbers of user and service groups and the regularization parameter $\tau$. The missing rate of $M$ is kept as 90%. These experiments are conducted on QoS_1 as the results from QoS_2 show a similar trend.

![Figure 1. Impact of the Numbers of User and Service Groups](image)

Figure 1 shows the impact of the numbers of user and service groups. To reduce parameter tuning, we always assume to have the same number of user and service groups, i.e., $k = l$. We vary the group number from 10 to 50 and an optimal MAE performance is achieved when $k = l = 45$. Small $k, l$ values lead to large user and service clusters, which may put together dissimilar users and services. On the other hand, overly increasing the number of groups will result in a large number of smaller user and service clusters, which will force to assign similar users into different clusters. The QoS prediction accuracy will be affected in both cases. It can be seen from the result that the MAE performance of TNR-MF is relatively robust to the group number. Good performance can be reported from a wide range of group numbers (i.e., from 20 to 50). This is due to the soft cluster membership mechanism as enforced by constraints in (4).

![Figure 2. Impact of the Regularization Parameter $\tau$](image)
Figure 2 shows the impact of the regularization parameter \( \tau \), which varies from 0.001 to 100. Similar to the group numbers, TNR-MF is also fairly robust to \( \tau \), which is justified through the stable MAE performance over a wide range of \( \tau \) values (i.e., from 0.001 to 1). It can also be seen from the figure that the MAE performance of SVT keeps improving with \( \tau \) increases. This is inline with the theoretical analysis from [3], which shows that SVT approaches a matrix with the lowest trace norm when \( \tau \to \infty \) and the obtained matrix is expected to well approximate the original low-rank matrix. However, SVT only focuses on discovering a low-rank matrix without considering the inherent clustered representation of QoS data. Hence, the best MAE performance achieved by TNR-MF (when \( \tau \) is around 0.1) clearly outperforms that of SVT (when \( \tau \) is at 100).

V. CONCLUSION

We develop a novel TNR-MF algorithm for accurate QoS prediction. The proposed algorithm incorporates both the low-rank structure and the clustered representation of real-world QoS data. It exploits trace norm regularized matrix factorization to seamlessly amalgamate these two features into a single unified objective function. A novel iterative algorithm is developed that employs simple gradient steps and efficient update rules to recover a QoS matrix from a small subset of observed QoS entries. Experimental results on two real-world QoS datasets and comparison with both neighborhood and model based collaborative filtering algorithms clearly justify the effectiveness of the proposed algorithm.

REFERENCES