WSP: A Network Coordinate based Web Service Positioning Framework for Response Time Prediction

Jieming Zhu†, Yu Kang†, Zibin Zheng†, and Michael R. Lyu†‡
†*Dept. of Computer Science and Engineering, The Chinese Univ. of Hong Kong, Hong Kong, China*
‡*School of Computer Science, National Univ. of Defence Technology, Changsha, China*
{*jmzhu, ykang, zbzheng, lyu*}*@cse.cuhk.edu.hk*

***Abstract*—With the rapid growth of Web services in recent years, the optimal service selection from functionally-equivalent service candidates has become more critical for building high quality service-oriented systems. To provide accurate QoS values for service selection, user-side QoS prediction thus becomes an important research problem. Although collaborative filtering based prediction approaches have been studied in several previous works, these methods suffer from the limitation of the sparsity of available historical QoS data, which greatly degrades the prediction accuracy. To address this problem, this paper proposes a Web service positioning (WSP) framework for response time prediction, which is one of the most important QoS properties. In our approach, a small set of landmarks are deployed to periodically monitor the response times of the Web service candidates and provide references to the numerous service users. By combining the advantages of network coordinate based approaches and collaborative filtering based approaches, the response times between users and Web services can be accurately predicted using their corresponding Euclidean distances. Extensive experiments are conducted based on our real-world QoS dataset collected on PlanetLab, comprising about 359,400 response time values from 200 users on 1,597 Web services. The experimental results show that our WSP approach outperforms the other existing approaches, especially when the historical data is sparse.**

***Keywords*-Web service positioning; QoS prediction; collaborative filtering; network coordinate**

I. Introduction

Web services are designed as computational components to build service-oriented distributed systems, such as e-commerce, automotive systems, multimedia services, etc [1]. With the rising popularity of Service-Oriented Architecture (SOA), more and more alternative Web services offered by different providers become available over the Internet to provide equivalent or similar functions for service users. With the growing number of Web services, it has become an urgent task to make effective selection from the large number of functionally-equivalent Web service candidates.

Quality-of-Service (QoS), such as response time, throughput, failure probability, reputation, etc., has been widely employed as a differentiating factor to describe and evaluate the non-functional characteristics of Web services [2]. QoS-based selection has become a promising approach for effective service selection and performance optimization. Among different QoS properties, response time is one of the most important properties, which stands for the time duration between user sending out a request and receiving a response. Due to the unpredictable network environment, users at different locations may experience different QoS performance of Web services, which cannot be identified by the Service Level Agreement (SLA) from the service providers' perspective. However, it is infeasible for each user to actively measure the QoS performance of all the Web services by invoking them due to the large number of service candidates in the Internet. As a result, efficient and effective QoS prediction approaches are urgently needed to provide accurate prediction of the QoS values of different Web services for each user without requiring real-world Web service invocations. In this paper, we will focus on response time prediction.

In recent literature, a number of QoS prediction approaches have been proposed [2], [3], [4], [5]. In these works, collaborative filtering (CF) is employed to make QoS predictions using Web service invocation histories of different users. With extensive experiments, these QoS prediction approaches have been shown to achieve good overall prediction accuracy under dense historical QoS data. However, the CF-based approaches suffer from a major limitation of the sparsity of the available historical QoS data. As reported in [2], given a very sparse response time matrix (where rows stand for users, columns represent Web services, and entries stand for the response time of a certain user on a certain Web service), the performance of CF-based approaches is greatly degraded. In practice, the response time matrix is very sparse since a user usually only invokes a few out of the numerous Web service candidates each time. Moreover, user-side response time performance of Web services is dynamically changing from time to time, influenced by the unpredictable network condition and the server workload. The out-of-date historical QoS data may not be able to provide accurate QoS prediction.

On the other hand, network coordinate systems (e.g., GNP [6]) are widely used in P2P networks to estimate the network distance between pairwise Internet hosts. The basic idea of network coordinate systems is to embed the Internet hosts into a high dimensional Euclidean space by assigning each host a coordinate in that space such that the measured

978-0-7695-4752-7/12 $26.00 © 2012 IEEE
DOI 10.1109/ICWS.2012.81

network distances between hosts can be well approximated by the corresponding Euclidean distances. After obtaining the coordinates of different hosts, we can use a simple Euclidean distance between two Internet hosts to predict the unknown network distance in constant time.

Inspired by the success of network coordinate based prediction approaches, and to address the data sparsity problem of CF-based approaches, we propose a Web service positioning (WSP) framework by combining the advantages of network coordinate based approaches and CF-based approaches. For ease of presentation, this paper only focuses on response time prediction. We will extend the proposed WSP framework to other QoS properties in our future work. In our WSP framework, firstly, we redesign the traditional GNP algorithm (typically employed in peer-to-peer scenario) to fit for the response time prediction of Web services in client-server scenario. Then, the available historical data of users (these data are employed in CF-based approaches for making prediction) are adopted to optimize the coordinate computation of users which enhances the prediction accuracy. Finally, extensive experiments are conducted based on real-world data. The comprehensive experimental results show that our WSP approach significantly outperforms the existing network coordinate based approaches and CF-based approaches.

The key contributions of this paper are two-fold:

- Firstly, we propose a Web service positioning (WSP) framework for response time prediction by combining advantages of network coordinate based approaches and CF-based approaches. Our WSP framework solves the data sparsity problem of CF-based approaches and significantly enhances the prediction accuracy.
- Secondly, we conduct a large-scale experiment to collect real-world response time data of Web services for verifying the performance of our WSP approach. The dataset includes 359,400 response time values from 200 users on 1,597 real-world Web services. To make our experiments reproducible and to promote future research, this reusable research dataset is publicly released online.

The rest of the paper is organized as follows. Section II introduces the related work. Section III presents the WSP Framework. Section IV describes our response time prediction algorithm in detail. Section V shows the experimental results. Finally, Section VI concludes the paper.

II. Related Work

A. Collaborative Filtering

Collaborative filtering (CF), as one of the most popular recommendation algorithms, has been widely used in commercial recommender systems, such as Amazon.com, Netflix.com, etc. In recent literature, collaborative filtering has been introduced to personalized QoS (e.g., response time) prediction for Web services [2], [3], [4], [5], [7], [8]. Shao et al. [3] introduced the user-based CF algorithm to predict the QoS values with similarity between users. Zheng et al. [2] proposed the item-based CF algorithm based on the similarity between Web services. These two algorithms are denoted as UPCC (user-based CF with PCC coefficient) and IPCC (item-based CF with PCC coefficient) respectively. In addition, a hybrid CF approach, UIPCC [2], was also proposed by combing UPCC and IPCC approaches. Experimental results have shown that CF-based approaches can achieve good overall prediction accuracy under dense historical QoS data. However, CF-based approaches suffer from the sparsity of available historical data in practice, which greatly reduces the prediction performance. In order to remit the data sparsity problem, Zhang et al. [9] put forward a time-aware CF approach, by adding time-dimension historical QoS data to their model, to enhance the QoS prediction accuracy. However, the accuracy improvement is limited. Especially, due to the high mobility of mobile users, no available historical QoS data can be obtained, which will cause the malfunction of CF-based approaches.

B. Network Coordinate System

On the other hand, the network coordinate system is proposed in [6] to estimate the network distances, i.e., round-trip time (RTT), between pairwise Internet hosts. Among various network coordinate systems, triangulated heuristic and global network positioning (GNP) are two widely employed approaches, due to their simplicity and generality.

Triangulated Heuristic [6] employs a kind of relative coordinates based on the triangle inequality. A fixed set of landmarks are deployed in the network as references. Then each ordinary host is assigned an n-tuple relative coordinate, composed of the network distances between the ordinary host and the landmarks. Given the relative coordinate of each host, we can obtain the upper bound U and the lower bound L of the network distance between two hosts by triangle inequality. The network distance can be estimated by the convex combination of U and L (e.g. $\frac{\mathrm{U}+L}{2}$). It is reported in [6] that taking the upper bound U as the network distance prediction result can achieve better performance. The triangulated heuristic approach is widely used in online shortest path distance prediction in large graphs [10].

GNP [6] is a typical landmark-based network coordinate system, which embed the Internet hosts into an Euclidean space for network distance estimation. After obtaining the coordinate of each host, the network distance between two Internet hosts can be well approximated by the corresponding Euclidean distance. Figure 1 illustrates a prototype of the network coordinate system. As we can see from the figure, the four Internet hosts can be embedded into a 2-dimensional Euclidean space by assigning each host a coordinate, and then the original network distances can obtain good estimation results using the corresponding Euclidean distances.

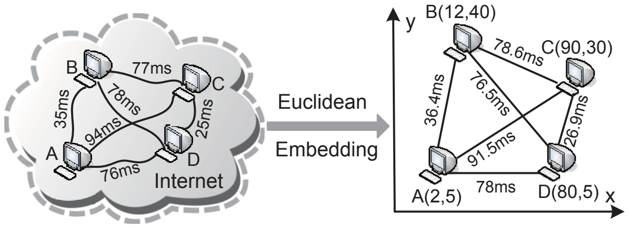


Figure 1. A Prototype of Network Coordinate System

Network coordinate systems, originally designed to estimate the network distances between Internet hosts in peer-to-peer distributed networks, have been comprehensively studied in recent years. To date, the adoption of network coordinate systems has benefited a variety of Internet applications, such as file sharing via Bit-Torrent [11], content distribution networks (CDN) [12], P2P multimedia streaming [13], etc. For further details on network coordinate systems, we refer the reader to a recent survey [14].

III. Web Service Positioning (WSP) Framework

To address the limitations of CF-based prediction approaches, we propose a Web service positioning framework (named WSP) to make response time prediction for Web services. In our framework, a small number of landmarks are deployed in the Internet to construct the network coordinate system and also to periodically monitor the available Web services. By adding the available historical data to the network coordinate model to optimize the coordinate computation, our proposed WSP approach combines the advantages of network coordinate based and CF-based approaches. As a result, our WSP approach not only can serve for users without available historical data (e.g., mobile users) but also enhances the prediction accuracy for users with sparse historical data.

The WSP framework is illustrated in Figure 2, which mainly includes the following procedures:

1) Offline Coordinates Updating: a) The deployed landmarks measure the network distances between each other (e.g. use ping to measure the RTT), and then construct a coordinate system by embedding the landmarks into an high-dimensional Euclidean space such that each landmark obtains a coordinate. *b)* The landmarks monitor the available Web services by periodically invoking. The coordinate of each Web service is obtained by taking the landmarks as references and embedding each Web service to the same coordinate system.

2) Online Web Service Selection: a) When a service user requests for a Web service invocation, it first measures the network distances to the landmarks (e.g by ping). Then the results are sent to a central node to compute the user's coordinate, also combining the available historical response time data. *b)* The response times between the user and all the available Web service candidates are easily predicted by computing the corresponding Euclidean distances via the coordinates. *c)* Optimal Web service is selected for the user based on the response time prediction results of Web service candidates. *d)* The user invokes the optimal Web service for composition, and also obtains the real response time data for this Web service. *e)* The response time database is updated with the new observation to contribute to the next Web service selection.

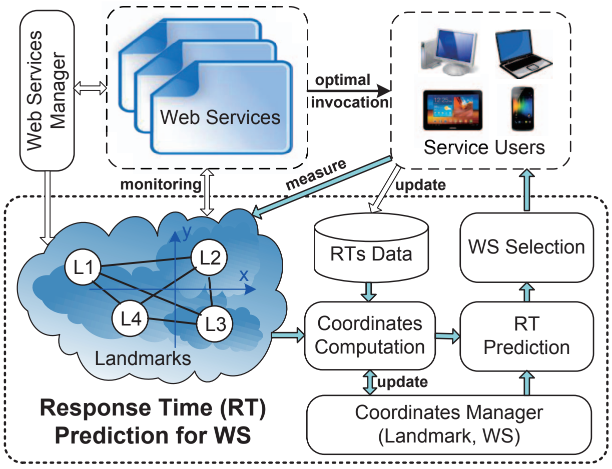


Figure 2. Web Service Positioning (WSP) Framework

IV. WSP-based QoS Prediction Algorithm

A. Landmark Coordinate Computation

Note that for an m-dimensional Euclidean space, we should use at least $m+1$ landmarks for coordinate computation, as it is impossible to construct an unique Euclidean space with less landmarks. It remains an open question about how to deploy the landmarks. In this paper, we select the landmarks from the candidate nodes using the spectral clustering based approach in [15]. In addition, we will study the impact of number of landmarks in Section V-E.

Suppose n landmarks, denoted by $L = \{l_i, i = 1, 2, \cdots, n\}$, are set up and deployed in the Internet. The network distances between landmarks are measured using ping messages, and then transmitted to a central node for coordinate computation, which can be expressed as an $n \times n$ distance matrix in the following:

$$D_L = \begin{bmatrix} 0 & d(l_1, l_2) & \cdots & d(l_1, l_n) \\ d(l_2, l_1) & 0 & \cdots & d_(l_2, l_n) \\ \vdots & \vdots & \cdots & \vdots \\ d(l_n, l_1) & d(l_n, l_2) & \cdots & 0 \end{bmatrix} \qquad (1)$$

where the entry $d(l_i, l_j)$ denotes the network distance between landmarks l_i and l_j. The distance matrix D_L is assumed to be symmetric along the diagonal, which is denoted as 0 where $l_i = l_j$.

Our goal is to embed these n landmarks into an m-dimensional Euclidean space R^m, such that each landmark obtains a coordinate, denoted as $x_{l_i} = (x^1_{l_i}, x^2_{l_i}, \cdots, x^m_{l_i})$, where $x^k_{l_i} \in R, 1 \leq k \leq m$. We define the objective function as the sum of error square in the following:

$$f_L(x_{l_1}, \cdots, x_{l_n}) = \sum_{l_i, l_j \in L, i>j} [\hat{d}(l_i, l_j) - d(l_i, l_j)]^2 \quad (2)$$

where $\hat{d}(l_i, l_j)$ denotes the predicted network distance between l_i and l_j as follows:

$$\hat{d}(l_i, l_j) = \|x_{l_i} - x_{l_j}\|_2 = \sqrt{\sum_{k=1}^{m} (x^k_{l_i} - x^k_{l_j})^2} \quad (3)$$

Directly minimizing Equation 2 will suffer from the overfitting problem, which means optimal solutions often lead to an accurate model with small errors on the landmarks embedding while having large errors on the unseen data, i.e., the unknown response times between users and Web services. To address this problem, we add a regularization term to penalize the norms of the solutions, expressed as follows:

$$\begin{aligned} f^{'}_L(x_{l_1}, \cdots, x_{l_n}, \lambda_l) &= \sum_{l_i, l_j \in L, i>j} [\hat{d}(l_i, l_j) - d(l_i, l_j)]^2 \\ &+ \lambda_l \sum_{i=1}^{n} \|x_{l_i}\|^2_2 \quad (4) \end{aligned}$$

Note that there are many solutions for minimizing the Equation 2, because any rotation or translation of the landmark coordinates will not influence the inter-landmark distances [6]. In addition to overcoming overfitting, the regularization term can also help to avoid the coordinate drift of the solution by choosing the coordinates with the smallest norm. The impact of the regularization term will be discussed in Section V-G.

With this formulation, the optimal coordinates of landmarks can be obtained by minimizing Equation 4, as a generic multi-dimensional global minimization problem. In this paper, we choose the *simplex downhill algorithm* [16] to solve this minimization problem.

Finally, the coordinates of the n landmarks are obtained and stored in the central node to provide references to the coordinate computation of Web services and users. Note that the coordinates of landmarks should keep updated periodically to track the changes of network conditions. For periodical re-computation, we can simply input the old coordinates as the start state each time, which can greatly accelerate the convergence of the minimization problem.

B. Web Service Coordinate Computation

In the WSP framework, a small number of landmarks monitor the available Web services by periodically invoking them. Suppose there are w available Web services, denoted by $S = \{s_i, i = 1, 2, \cdots, w\}$. Then an $n \times w$ matrix composed of network distances between n landmarks and w Web services can be obtained, expressed as follows:

$$D_{LS} = \begin{bmatrix} d(l_1, s_1) & d(l_1, s_2) & \cdots & d(l_1, s_w) \\ d(l_2, s_1) & d(l_2, s_2) & \cdots & d(l_2, s_w) \\ \vdots & \vdots & \cdots & \vdots \\ d(l_n, s_1) & d(l_n, s_2) & \cdots & d(l_n, s_w) \end{bmatrix} \quad (5)$$

where the entry $d(l_i, s_j)$ denotes the network distance between landmark l_i and Web service s_j.

The network distances to Web services are measured by landmarks and then transmitted to a central node to compute the coordinate for each Web service. Therefore, each Web service is embedded into the network coordinate system by taking the coordinates of the landmarks as references. Given a Web service $s_j (1 \leq j \leq w)$, the m-dimensional coordinate x_{s_j} can be obtained by minimizing the following objective function.

$$f_S(x_{s_j}, \lambda_s) = \sum_{l_i \in L} [\hat{d}(l_i, s_j) - d(l_i, s_j)]^2 + \lambda_s \left\|x_{s_j}\right\|^2_2 \quad (6)$$

where $\hat{d}(l_i, s_j)$ denotes the predicted network distance between landmark l_i and Web service s_j. And also $\lambda \left\|x_{s_j}\right\|^2_2$ is the regularization term.

Similarly, this computation can also be cast as a generic multi-dimensional global minimization problem such that we can solve it with *simplex downhill algorithm*. Meanwhile the coordinates of the available Web services are updated periodically.

C. Service User Coordinate Computation

Any service user can request our WSP system for optimal Web service selection. At the beginning, the user measures the network distances to the landmarks using ping messages, and then transmit the results to the central node for coordinate computation. The measured network distances can be denoted as a vector in the following:

$$D_{uL} = [d(u, l_1), d(u, l_2), \cdots, d(u, l_n)] \quad (7)$$

where $d(u, l_i)$ denotes the network distance between the user u and the landmark l_i.

To enhance the prediction accuracy, we also incorporate the advantage of CF-based approaches by making effective use of the available historical data. Suppose the available response time data between the user u and the Web services is denoted as $\{d(u, s_i), s_i \in S_A\}$, where S_A is the Web service set with available historical data, then we propose to minimize the following objective function to obtain the coordinate of the user, i.e. x_u.

$$\begin{aligned} f_u(x_u, \lambda_u) &= \sum_{l_i \in L} [\hat{d}(u, l_i) - d(u, l_i)]^2 \quad (8) \\ &+ \sum_{s_i \in S_A} [\hat{d}(u, s_i) - d(u, s_i)]^2 + \lambda_u \|x_u\|^2_2 \end{aligned}$$

where $\hat{d}(u, l_i)$ denotes the predicted network distance between user u and landmark l_i, and $\hat{d}(u, s_i)$ denotes the predicted response time value between user u and the Web service s_i in S_A. The first part of the objective function $f_u(x_u, \lambda_u)$ employs the reference information of landmarks while the second part takes advantage of the available historical data. Besides, the regularization term is also considered in the third part.

D. Response Time Prediction and Web Service Selection

After obtaining the coordinate of the user, as well as the coordinates of all the monitored Web services, the response time prediction can be easily achieved by taking the Euclidean distance between the coordinates as follows:

$$\hat{d}(u, s_i) = \|x_u - x_{s_i}\|_2, \quad s_i \in S, s_i \notin S_A \tag{9}$$

where $\hat{d}(u, s_i)$ denotes the prediction value between user u and Web service s_i. $s_i \in S$, $s_i \notin S_A$ means the set of Web services with unknown response time data.

With all the response time predictions, the optimal Web service can be selected for the user according to the response time performance. However, the specific QoS-aware service selection approach for Web service composition is out of the scope of this paper.

V. Experiments

A. Data Collection

To evaluate the prediction performance, real-world Web service dataset of response time is needed. Although several QoS datasets for Web services have been collected in the previous work [2], they are not applicable for our experiment due to the lack of network distances among landmarks to construct the network coordinate system.

In this paper, we collect a new QoS dataset for our experiment, comprising the response times between 200 users (PlanetLab nodes) and 1,597 Web services, together with the network distances between the 200 distributed nodes, for our WSP approach.

To collect the real-world data of Web services, we have an access to the open research platform, Planetlab[1], which is widely used for experiments of distributed systems and networks. We first get a list of 588 active PlanetLab nodes via CoMon[2] service, since some nodes may shut down or lose connection of the Internet. Meanwhile, about 5,800 Web services are obtained by crawling Web service information from the Internet.

To obtain the response time data, we use ping messages to measure the round-trip time (RTT) from each PlanetLab node to each Web service, assuming that the service-running time is equivalent for each Web service due to their same function. We send 32-byte ping packets continually for ten times and take the average RTT from all replies as the response time. Similarly, the network distances among the PlanetLab nodes are obtained as well. The raw data is then post-processed to retain the nodes and Web services that are all reachable. Finally, we are left with 200 PlanetLab nodes and 1,597 Web services. The relatively low yield is partially due to the case that some Internet hosts are ping unavailable, and partially due to the failure of the Internet connection. Consequently, a 200-by-1597 matrix of response times and a 200-by-200 matrix of network distances are obtained.

Table I
Descriptions of WS response time Dataset

Statistics	Values
Num. of records	359,400
Num. of service users	200
Num. of Web services	1,597
Minimum response time	0.008 ms
Maximum response time	2,976.714 ms
Mean of response time	71.984 ms
Standard deviation of response time	64.746 ms

The statistics of our QoS dataset is summarized in Table I. Our dataset is also publicly released online[3] for future research.

B. Evaluation Metrics

In our experiment, we employ two metrics, *Mean Absolute Error (MAE)* [9] and *Median Relative Error (MRE)*, to evaluate the prediction performance of our proposed WSP approach. The two metrics are defined as follows:

- *MAE*: This metric is widely employed to measure the average prediction accuracy.

$$MAE = \frac{\sum_{i,j} \left| \hat{d}(u_i, s_j) - d(u_i, s_j) \right|}{N} \tag{10}$$

  where $\hat{d}(u_i, s_j)$ and $d(u_i, s_j)$ denote the predicted value and the measured value, respectively, between service user u_i and Web service s_j. N is the number of predicted items.
- *MRE*: This metric is median value of all the relative error values.

$$MRE = \underset{i,j}{Median} \frac{\left| \hat{d}(u_i, s_j) - d(u_i, s_j) \right|}{d(u_i, s_j)} \tag{11}$$

  which means 50% of the relative errors are below *MRE*.

C. Performance Comparison

In this section, in order to evaluate the prediction accuracy of our proposed WSP approach, we compare our method with other existing approaches in the following:

[1] http://www.planet-lab.org

[2] http://comon.cs.princeton.edu

[3] http://www.wsdream.net

Table II
PERFORMANCE COMPARISON (SMALLER MAE, RMSE OR MRE VALUES MEANS BETTER PERFORMANCE)

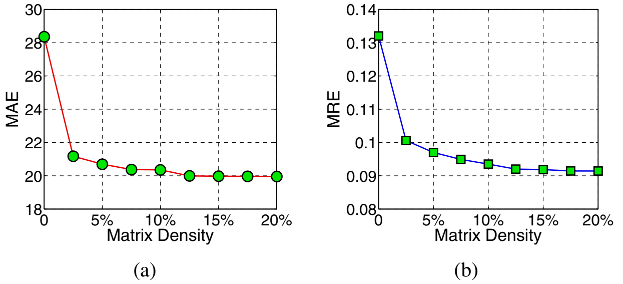
Methods	Density = 0		Density = 5%		Density = 10%		Density = 15%	
	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE
UPCC	N/A	N/A	33.4439	0.1842	25.4751	0.1329	21.2634	0.1051
IPCC	N/A	N/A	51.9462	0.3064	31.3841	0.1674	26.2708	0.1335
UIPCC	N/A	N/A	33.7476	0.1856	**25.3795**	**0.1317**	**21.0852**	**0.1027**
Triangulated	33.9315	0.1733	33.9315	0.1733	33.9315	0.1733	33.9315	0.1733
GNP	**29.2793**	**0.1375**	**29.2793**	**0.1375**	29.2793	0.1375	29.2793	0.1375
WSP	**28.3502**	**0.1316**	**20.6982**	**0.0972**	**20.3531**	**0.0927**	**19.9709**	**0.0922**
Improvements(%)	3.17%	4.29%	29.31%	29.31%	19.80%	29.61%	5.28%	10.22%

1) **UPCC**: This method is first introduced to Web service QoS prediction in [3], which employs the similarity between users to predict the response time values.
2) **IPCC**: This method employs the similarity between Web service items for Web service QoS prediction.
3) **UIPCC**: This is a hybrid method, proposed in [2], by combing both user-based and item-based collaborative filtering approaches, which can make full use of the similarity between users and the similarity between Web service items.
4) **Triangulated heuristic**: This method is based on the triangle inequality in the metric space. In this paper, we take the the upper bound as the response time prediction result as indicated in Section II-B.
5) **GNP**: GNP is proposed as a landmark-based network coordinate system to estimate network distances between Internet hosts for P2P networks in [6].

In this experiment, we choose 16 nodes as landmarks from our dataset as did in GNP [6] (the impact of the number of landmarks will be discussed in Section V-E), while the remaining 184 computer nodes are taken as service users. Therefore, the measured response time data between 184 service users and 1,597 Web services can be denoted as a 184 × 1597 matrix. As we mentioned in the previous section, the available historical data is very sparse. In order to simulate the sparse situation in real world, we randomly remove entries from the data matrix such that each user only keeps a few available historical values. In this way, we vary the matrix density as 0, 5%, 10%, 15%. Particularly, matrix density $= 0$ means no historical data of users are employed, such as for mobile users, whose historical data may vary significantly due to their high mobility and are not applicable for response time prediction. Matrix density $= 5\%$, for example, indicates that each user has 5% (i.e. about 80) response time data out of all the Web services. The removed entries are used as the expected values to verify the prediction quality. In the sequel, for simplicity, we set $\lambda_l = \lambda_s = \lambda_u = \lambda$, and denote n as the number of landmarks, m as the coordinate dimensionality. In this experiment, the parameter settings are $\lambda = 0.1, m = 10$. Each approach is performed 100 times and the average values are reported. In contrast, we set Top-K$= 10$, $\lambda = 0.1$ for CF-based approaches [2]. The prediction performance of different approaches under two metrics are shown in Table II.

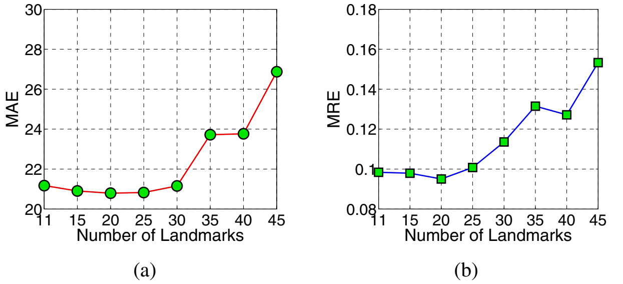
The experimental results show that:

- Our WSP approach obtains smaller MAE and MRE values consistently under different matrix densities, which indicates that our approach outperforms the others. The last row shows the percentages of the accuracy improvements of our WSP approach, compared with the best of other existing methods.
- While CF-based approaches are heavily influenced by the matrix density, our WSP approach is less sensitive to the matrix density and obtains good prediction accuracy even under sparse historical data. For instance, the WSP has about 20% improvement compared with the UIPCC method, with 10% historical data.
- Especially, for matrix density $= 0$, the CF-based approaches (UPCC, IPCC and UIPCC) run into a malfunction (denoted as N/A) while landmark-based approaches (Triangluated, GNP, and WSP) achieve good overall prediction accuracy. It implies that our WSP approach can also serve well for newly joining users or mobile users without available historical data.
- Our WSP approach outperforms the traditional network coordinate based approaches, triangulated heuristic and GNP, even at matrix density $= 0$, indicating that our WSP approach makes improvement based on GNP.
- The IPCC approach performs worse than the UPCC approach in our experiments. This is because the IPCC method cannot find enough similar neighbors as the number of users is much smaller than the number of Web services. In other words, it is the data sparsity that significantly degrades the performance of the IPCC method.
- With the increase of matrix density, the triangulated heuristic approach and GNP approach have no performance improvement since they make no use of the historical data.



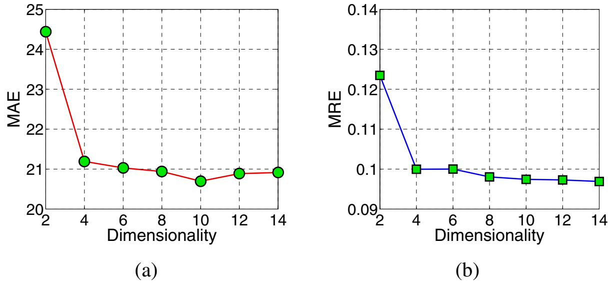
(a) (b)

Figure 3. Impact of the Matrix Density



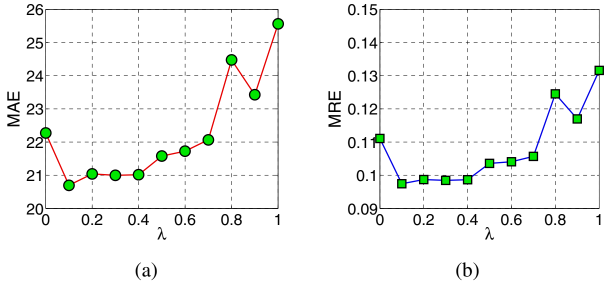
(a) (b)

Figure 4. Impact of the Number of Landmarks



(a) (b)

Figure 5. Impact of the Coordinate Dimensionality



(a) (b)

Figure 6. Impact of the Regularization Term

To sum up, our proposed WSP approach combines the advantages of network coordinate based approaches and CF-based approaches, and achieves better performance compared with the existing prediction methods.

D. Impact of the Matrix Density

To study the impact of the matrix density on the prediction accuracy, we vary the matrix density from 0 to 20% at the step of 2.5%, where matrix density $= 0$ means making response time predictions without using historical data, i.e. $S_A = \emptyset$ in Equation 8. In this experiment, without loss of generality, we set $n = 16$, $m = 10$, and $\lambda = 0.1$.

The experimental results are shown in Figure 3, composed of MAE and MRE values under different matrix densities. We can observe that dense historical data can benefit the prediction performance. However, after significantly decreasing when the matrix density varies from 0 to 2.5%, MAE and MRE both keep steady with a little reduce when the matrix density becomes denser. In other words, the prediction performance of WSP approach is less sensitive to the sparsity of data matrix, as a result, addresses the limitations of practical data sparsity problem of CF-based approaches.

E. Impact of the Number of Landmarks

The landmark deployment (e.g., the position and number) is very essential to the performance of our WSP approach. In this experiment, we select the landmarks from the candidate nodes using the spectral clustering based approach in [15]. To characterize the impact of the number of landmarks, we conduct the experiment by varying the number of landmarks from 11 to 45. We also set $m = 10$, $\lambda = 0.1$ and matrix density $= 5\%$. Note that there should be more than 11 landmarks for 10-dimensional Euclidean space construction. The results of MAE and MRE are illustrated in Figure 4. We can observe that the MAE and MRE values both decrease slightly when the number is less than 20, and then rise when the number is larger than 25. We can find that the large number of landmarks may not make for the prediction performance improvement, since there exist larger errors when embedding the Web services and users into Euclidean space with too many reference nodes, as a result of the triangle inequality violations (TIV) of network latencies [15]. In practice, we can deploy enough landmarks while each Web service and user only choose n landmarks as references, which can also avoid the single point of failure of landmarks.

F. Impact of the Coordinate Dimensionality

Dimensionality is a key factor when embedding the Internet hosts into an Euclidean space. We may wonder how many dimensions should be used to construct the coordinate system in WSP. Intuitively, higher dimensionality contributes to more accurate coordinate computation. To characterize the impact of the dimensionality, we conduct experiments with our dataset and vary the dimensionality from 2 to 14 at the step of 2. We also set $n = 16, \lambda = 0.1$ and matrix density$= 5\%$. The experimental results are illustrated in Figure 5. We can observe that both MAE and MRE values decrease with the increase of the dimensionality of coordinates, but the accuracy improvement diminishes when the dimensionality is larger than 8. As a result, higher dimensionality beyond a certain point only makes little performance improvement.

G. Impact of the Regularization Term

To address the overfitting problem when computing coordinates, we introduce a regularization term to penalize the norms of the solutions which is widely adopted in machine learning area. In addition, the regularization term can also avoid the coordinate drift due to the non-uniqueness of the solution by choosing the coordinates with the smallest norm.

In this experiment, we vary the λ from 0 to 1, while $\lambda = 0$ means no regularization term is used. For other parameters, we set $n = 16, m = 10$, and matrix density $= 5\%$. The experimental results are shown in Figure 6. As is shown in the figure, when $\lambda = 0.1$, smaller MAE and MRE values are obtained compared with $\lambda = 0$, indicating that the regularization term can contribute to the prediction performance improvement. However, MAE and MRE rise with the increase of λ. Therefore, the prediction performance is sensitive to λ and we set $\lambda = 0.1$ in this paper, which is verified to achieve a good performance.

VI. Conclusion and Future Work

In this paper, we propose a network coordinate based Web service positioning framework for response time prediction. By combining the advantages of network coordinate based approaches and CF-based approaches, our WSP framework is constructed to support online personalized response time prediction for service users. Extensive experimental results show that our proposed WSP approach solves the data sparsity problem of CF-based approaches and significantly enhances the prediction accuracy. Besides, our WSP approach can also serve for users without available historical data, such as mobile users and the newly joining users, which is not applicable for CF-based approaches.

This paper focuses on the response time prediction of Web services. In the future, we will extend our WSP framework to other QoS properties, and investigate QoS-based service selection approaches by taking into account these available QoS values. We will further study to improve the accuracy of Euclidean embedding by taming the triangle inequality violations in the Internet. In addition, we will deploy practical systems to evaluate our WSP framework more realistically.

Acknowledgment

The work described in this paper was fully supported by the Research Grants Council of Hong Kong, China (Project No. CUHK 415311), the NSFC/RGC Joint Research Scheme sponsored by the Research Grants Council of Hong Kong and the National Natural Science Foundation of China (Project No. N_CUHK405/11), the National Natural Science Foundation of China under Grant No. 61100078, and the National Basic Research Program of China (973) under Grant No. 2011CB302603.

References

[1] L. Zhang, J. Zhang, and H. Cai, *Services Computing: Core Enabling Technology of the Modern Services Industry*. Tsinghua University Press, 2007.

[2] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," *IEEE Trans. on Services Computing*, vol. 4, no. 2, pp. 140–152, 2011.

[3] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for web services via collaborative filtering," in *Proc. of the IEEE ICWS'07*, 2007, pp. 439 –446.

[4] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized QoS-aware web service recommendation and visualization," *IEEE Trans. on Services Computing*, accepted, 2011.

[5] Y. Jiang, J. Liu, M. Tang, and X. F. Liu, "An effective web service recommendation method based on personalized collaborative filtering," in *Proc. of the IEEE ICWS'11*, 2011.

[6] T. S. E. Ng and H. Zhang, "Predicting internet network distance with coordinates-based approaches," in *Proc. of the IEEE INFOCOM'02*, 2002.

[7] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based web service recommender system," in *Proc. of the IEEE ICWS'09*, 2009, pp. 437–444.

[8] Q. Xie, K. Wu, and J. Xu, "Dynamic selection of web services based on collaborative filtering," *Advances in Information Sciences and Service Sciences (AISS)*, vol. 3, no. 11, pp. 374–381, 2011.

[9] Y. Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for web services," in *Proc. of the IEEE ISSRE'11*, 2011, pp. 210–219.

[10] M. Qiao, H. Cheng, and J. X. Yu, "Querying shortest path distance with bounded errors in large graphs," in *Proc. of the SSDBM'11*, 2011, pp. 255–273.

[11] M. Steiner and E. W. Biersack, "Where is my peer? evaluation of the vivaldi network coordinate system in azureus," in *Proc. of Networking'09*, 2009, pp. 145–156.

[12] N. Ball and P. Pietzuch, "Distributed content delivery using load-aware network coordinates," in *Proc. of the ACM CoNEXT'08*, 2008, pp. 77:1–77:6.

[13] S. Lee and S. Sahu, "Network distance based coordinate systems for p2p multimedia streaming," in *Proc. of the IEEE/IFIP NOMS'10*, 2010, pp. 793–796.

[14] B. Donnet, B. Gueye, and M. A. Kâafar, "A survey on network coordinates systems, design, and security," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 4, pp. 488–503, 2010.

[15] Y. Mao, L. Saul, and J. Smith, "IDES: An internet distance estimation service for large networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2273 –2284, 2006.

[16] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.