A Mini-Project Report

On

A Spatial-Temporal QoS Prediction Approach for Time-aware Web Service Recommendation

Submitted by

Aparna P L (14IT132)

Padmaja B(14IT128)

Pooja M Soundalgekar (14IT230)

VIII Sem B.Tech (IT)

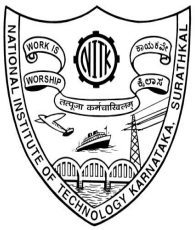
in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY



Department of Information Technology

National Institute of Technology Karnataka, Surathkal

Dec - May 2017 - 2018

CERTIFICATE

This is to certify that the project entitled "A Spatial-Temporal QoS Prediction Approach for Time-aware Web Service Recommendation" is a bonafide work carried out as a part of the course **Web Services (IT450)**, under my guidance by

1. Aparna P L(14IT132)
2. Padmaja B(14IT128)
3. Pooja M Soundalgekar (14IT230)

students of VIII Sem B.Tech (IT) at the Department of Information Technology, National Institute of Technology Karnataka, Surathkal, during the academic year Dec - May 2017-18, in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology, at NITK Surathkal.

Place: _____

Signature of the Instructor

Date:

Abstract

Due to the popularity of service-oriented architectures for various distributed systems, an increasing number of Web services have been deployed all over the world. Recently, Web service recommendation became a hot research topic, one that aims to accurately predict the quality of functional satisfactory services for each end user. Generally, the performance of Web service changes over time due to variations of service status and network conditions. Quality of service (QoS) is widely employed for describing nonfunctional characteristics of web services. Instead of employing the conventional temporal models, a novel spatial-temporal QoS prediction approach for time-aware Web service recommendation has been implemented, where a sparse representation is employed to model QoS variations. Specifically, zero-mean Laplace prior distribution assumption on the residuals of the QoS prediction has been carried out, which corresponds to a Lasso regression problem. To effectively select the nearest neighbor for the sparse representation of temporal QoS values, the geolocation of web service is employed to reduce searching range while improving prediction accuracy.

DECLARATION

We hereby declare that the project entitled "A Spatial-Temporal QoS Prediction Approach for Time-aware Web Service Recommendation" submitted as part of the partial course requirements for the course **Web Services (IT450)** for the award of the degree of Bachelor of Technology in Information Technology at NITK Surathkal during the Dec-May 2017-18 semester has been carried out by us. We declare that the project has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles elsewhere.

Further, we declare that we will not share, re-submit or publish the code, idea, framework and/or any publication that may arise out of this work for academic or profit purposes without obtaining the prior written consent of the Course Instructor.

Name and signature of the Students:

1. Aparna P L(14IT132)
2. Padmaja B(14IT128)
3. Pooja M Soundalgekar (14IT230)

Place:

Date:

Contents

List of Figures

1 Introduction

With the development of Service-oriented architecture (SOA), Web services have become standard software components deployed in the Internet. When developing service-oriented applications, designers try to find and reuse existing services to build the system business process. Meanwhile, multiple Web services with the same functionality are offered by different service providers. Users pick over the services based on their Quality of Service (QoS), such as price, availability and reputation. [1]

Practically, the end users are from various geographical locations whose QoS values are greatly dependent on network conditions and network geographical locations. Different users may have quite different QoS even using the same service, and the QoS values evaluated by one user cannot be used directly by others. This makes QoS prediction a challenging task [Chen et al. 2013]. To tackle this issue, collaborative filtering is widely employed to predict the QoS values of all candidate services for the personalized Web service recommendation.

In a highly dynamic Internet environment, QoS values for Web services usually change with time [Zheng et al. 2012], and service status, such as the number of clients and network conditions, always varies with time [Zhang et al. 2011]. Therefore, Web services with optimal QoS values also change over time, sometimes drastically. For a better illustration, we employ the following function to evaluate the changing rate of QoS values between two adjacent time slots [Zheng et al. 2012]:

$$r_i = |q_i - q_{i-1}|/q_{i-1} \tag{1}$$

where qi and qi-1 represent the QoS values of the time slots i and i 1 respectively. ri denotes the changing rate between these two time slots. If r i is greater than a threshold , we define qi as a sudden change in QoS value. We select a set of temporal response time data from a QoS repository [Zhang et al. 2011] that includes the QoS values from a user of two different services.

Generally, data-driven methods can be employed for temporal QoS prediction, and these are usually formulated as missing value problems. Specifically, a large number of user-service-time-aware QoS values are collected offline and are further used to build a tensor representation. Assuming that the QoS values lie in few latent subspaces, the missing item is predicted by taking advantage of tensor decomposition [Zhang et al. 2011].

However, this low rank assumption may not be valid for those temporal QoS values with a large amount of sudden changes. In the time-aware service recommendation, a number of QoS values for candidate services should be predicted for the current time. To account for the dynamic environment of Web service recommendation, the QoS prediction should be invoked continuously to take advantage of currently and recently collected QoS values from other users. Factorization-based methods require rebuilding the model in order to make use of the newly accumulated QoS information. This will incur heavy computational cost.

To address these limitations, we propose a spatial-temporal QoS prediction approach for time-aware Web service recommendation in which sparse representation is employed to model QoS variations. Specifically, we make a zero-mean Laplace prior distribution assumption on the residuals of QoS prediction, which corresponds to a Least Absolute Shrinkage and Selection Operator (Lasso) problem. To effectively select the nearest neighbor for building the sparse representation, the geo-locations of Web services and users are employed to reduce the searching range while improving prediction accuracy.

1.1 *Motivation*

It is important to study personalized Web service Recommendation as evaluating all the real-world candidate services with equivalent function in order to find the optimal one is quite ineffective. The key to service selection and recommendation is Quality of Services (QoS), which is designed to distinguish among different functionally equivalent services. Specifically, QoS is defined as a set of user experienced properties for a specific Web service including response time, throughput, reputation, and the like. QoS prediction is essential to effective personalized service recommendation.

2 Literature Survey

The key to QoS-aware Web service recommendation is to accurately predict QoS values [3][4][5], which are employed to help the user distinguish among a huge number of functionally equivalent Web services and improve the overall quality of Web applications.

In general, QoS performance for Web services is typically measured from the user side, including response time, reliability, and throughput. Therefore, different users may have quite different QoS values for the same service. In this project, we focus on the time-aware Web service recommendation. The related work can be roughly categorized into two groups: temporal model-based methods and data-driven approaches[2].

For temporal model-based methods, the collected QoS values are typically treated as a time series, in which lots of techniques on modeling the dynamic behaviors of QoS characteristics can be employed to predict the temporal QoS values. Godse et al. [2010][6] monitor the QoS data continuously while forecasting the QoS values based on an ARIMA model, which involves a single time expert without human intervention during the execution. Li et al. [2009][7] take advantage of the structural equation time series model to fit the QoS values of Web services and to predict the change of QoS values dynamically. However, this approach mainly concerns global QoS containing multiple QoS attributes, which does not focus on real-world QoS values (i.e., response time and throughput). Cavallo et al. [2010][8] conducted an empirical study on QoS prediction including average values, linear models, and time series that demonstrated that the time series-based method can achieve promising prediction accuracy. Zeng et al. [2008][9] argue that the performance metrics and Key Performance Indicators (KPI) can be predicted using the ARIMA time series model in QoS management. They present the design and implementation of a novel event-driven QoS prediction system that can process operational service events in a real-time fashion to predict or refine the prediction of metrics and KPIs. Amin et al. [2012][10] integrate both ARIMA and GARCH models to capture QoS attributes volatility. Hooman and Kennedy [2009][11] propose a Historical Symbolic Delay Approximation (HDAX) model to predict network delays. Experimental results demonstrate that their method shows better prediction accuracy in forecasting the delay-time series as well as in reducing the time cost of the forecasting method. The main show-stopper is that these methods require a large amount of historical QoS values and use some special assumptions.

For the data-driven approaches, QoS value prediction is usually formulated as a missing item problem, which is typically solved by Collaborative Filtering (CF) and factor analysis methods. CF was introduced to personalized QoS prediction and Web service recommendation [Chen et al. 2013; Shao et al. 2007; Zheng et al. 2009, 2011; Sun et al. 2012; Luo et al. 2012], which can be categorized into two groups: memory-based methods and model-based approaches. The memory-based CF methods employ user-rating data to compute the similarity between users or items and predict QoS values accordingly [McLaughlin and Herlocker 2004; Miller et al. 2003][12]. They include user-based approaches, item-based approaches, and hybrid approaches. Tang et al. [2012] [13] propose the location-aware DF method that incorporates the locations of both users and services, which reduces the search space for similar users and services and improves Web service recommendation performance. Sun et al. [2012] [14] propose a normal recovery CF approach with improved Web service similarity. Yao et al. [2013] [15] combine CF and content-based recommendation by taking into consideration both rating data and the content of Web services.

All these approaches, however, neglect QoS variations across different times. The model-based CF approaches, such as Bayesian model and K-means clustering, learn the statistical model from a training dataset, including clustering models [Ungar and Foster 1998], latent factor models, and an aspect model. To improve prediction accuracy, Luo et al. [2012] [16] propose a collaborative QoS prediction framework with Location-Based Regularization (LBR), which incorporates geographical information to identify the neighborhood using the local connections between Web services users.

2.1 *Outcome of Literature Survey*

Most of the approaches, however, neglect QoS variations across different times. To address these limitations, we propose a spatial-temporal QoS prediction approach for time-aware Web service recommendation in which sparse representation is employed to model QoS variations. Specifically, we make a zero-mean Laplace prior distribution assumption on the residuals of QoS prediction, which corresponds to a Least Absolute Shrinkage and Selection Operator (Lasso) problem. To effectively select the nearest neighbor for building the sparse representation, the geo-locations of Web services and users are employed to reduce the searching range while improving prediction accuracy.

2.2 *Problem Statement*

A Spatial-Temporal QoS Prediction Approach for Time-aware Web Service Recommendation using LASSO technique.

2.3 *Objectives*

1. Comparison between general regressor and Lasso regressor for time-aware prediction
2. Using geolocation information like latitude, longitude for QoS prediction
3. Selecting user-web service pairs based on cross-correlation and geodesic distance measure.

3 Methodology and Framework

3.1 *System Architecture*

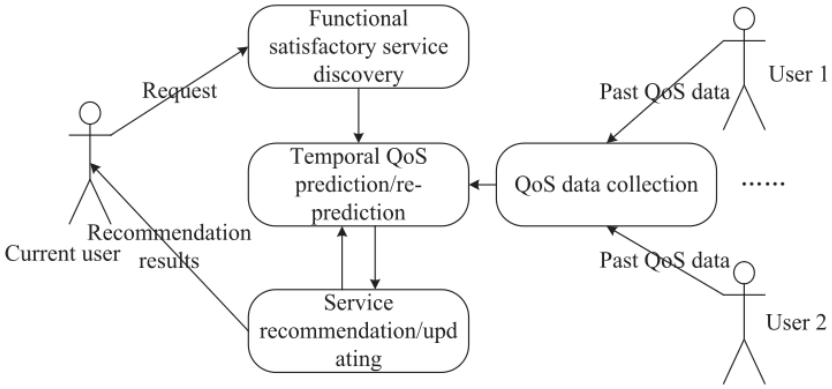


Figure 3.1: System Architecture for A Spatial-Temporal QoS Prediction Approach for Time-aware Web Service Recommendation

The objective of the time-aware service recommendation is to provide functionally optimal services to each user for the current time slot (i.e., an hour or a month). To improve recommendation accuracy, service users usually require an evaluation of the performance of all the functionally satisfied services for the current time slot, which incurs heavy computational cost and network traffic. To solve this problem, we present an effective spatial-temporal QoS prediction for time-aware service recommendation. As shown in Figure 1, the overall procedures of the time-aware Web service recommendation are summarized in the following: (1) An end user requests a service in the current time slot; (2) A set of satisfactory functional services are retrieved by the service search module; (3) The temporal QoS prediction / re-prediction module predicts the QoS values of the set of retrieved service for the current time slot; (4) The service recommendation module selects the top m services with the optimal predicted QoS values for the current time slot and recommends these services to the end user.

As described in Zheng et al. [2011], the QoS data are collected through a user collaborative QoS collection mechanism that archives the current or most recent QoS values from different services continuously contributed by different service users over time. Based on the newly collected QoS values, the temporal QoS prediction can be performed continuously. Additionally, repredicted QoS can be conducted using the newly collected temporal

QoS values to further improve the prediction accuracy of the recommended services for the current time slot since more and more temporal QoS values are accumulated in the current time slot. Therefore, the services with optimal predicted QoS values for the current time slot can be adapted, and the recommended services can also be updated continuously to the service user. The system architecture of the same is given in Figure 3.1.

3.2 *Temporal QoS Prediction by Generic Regression*

The key to time-aware Web service recommendation is to predict QoS values from a set of collected temporal QoS data. As with other prediction problem, QoS prediction can be solved using data driven methods, such as linear regression. The fundamental idea of the data-driven approach is to represent the QoS value by a linear combination of a set of similar examples. The popular matrix factorization-based CF methods also assume that the QoS data lies on the subspace spanned by the historical training data, which also resorts to representing the data point using the linear combination of similar examples. To this end, we formulate QoS prediction as a generic regression problem.

Given a set of collected temporal QoS data X with respect to users and services, we aim to predict the QoS value y n at time slot n for the user u and service v. Let y = (y1, . . . , yn-1) T denote the collected temporal QoS data for the user u and service v, where yk (1 k n 1) is the collected QoS value at time slot tk. If the QoS value at tk is not collected, then yk is set to null (invalid). The fundamental idea of linear regression for QoS prediction is that the current QoS value can be represented by the linear combination of K most similar QoS sequences X = (x1 , . . . , xk) in the dataset. Therefore, we try to find a latent mapping function f (x) between the previously collected historical data and the observation of QoS value for the user u and service v. Typically, we can assume that the QoS value y differs from the prediction by some additive Gaussian noise with zero mean and variance 2 . Therefore, the target QoS value y can be represented as y= f(x) + ϵ

Let w $\in$ R^{K*1} denote a vector of linear combination coefficients; then, the mapping function f (x) can be formulated as:

$$f(x) = \sum_{K}^{i=1} W_i X_i = W_x \tag{2}$$

Least squares regression minimizes the sum of squared distances between the observed QoS values for the user u and service v and the one predicted by the linear mapping function by minimizing the squared residual error:

$$\min_{w} = \|y - w^T X\|_2^2 + \lambda_1 \|w\|_2^2 \tag{3}$$

where $||w||^2$ is the regularization term and λ_1 is the regularization coefficient to avoid the overfitting issue. Equation (3) has the following closed-form solution.

3.3 *QoS Prediction Using Lasso*

In this project, we consider the zero mean Laplace distribution with probability density functions that have abrupt changes in gradient, which corresponds to a Lasso problem. Lasso shrinks some coefficients and sets others to zero, and hence tries to retain the good features of both subset selection and regularized least square regression. This further leads to an effective sparse representation for temporal QoS data. Instead of L2 norm regularization in least square regression, Lasso imposes an L1 penalty on the linear combination coefficients, which leads to the following optimization problem:

$$\min_{w} = \|y - w^T X\|_2^2 + \lambda_1 \|w\|_1 \tag{4}$$

where $\lambda \succeq 0$ is the regularization coefficient that controls the amount of shrinkage on the predictions. Therefore, some coefficients shrink and may be exactly equal to zero. Obviously, this Lasso minimization problem uses a quadratic programming with linear inequality constraints, which can be efficiently solved by either a Least Angle Regression (LARS) algorithm or an L_1 -regularized least squares method.

Once the optimal linear combination coefficients w is obtained through Lasso, the QoS value for the current time slot can be directly predicted by Equation (2). Note that the computational cost for Lasso is closely related to the size of selected example QoS sequences K. Because there are usually many user-service pairs, it is very time consuming to use all the related information to predict the QoS value from a temporal sequence. Practically, we only select a very small portion of examples from the collected QoS dataset in order to facilitate the online applications. In the following, we discuss how to effectively choose the most similar sequences in the time-aware Web service recommendation task.

3.4 *Spatial-Temporal QoS Prediction*

To capture the dynamic characteristics of the input sequence y, we employ the normalized cross-correlation between each sample x in the collected QoS dataset with y as the similarity measure:

$$S(x, y) = (x - x^{-})(y - y^{-})/\|x - x^{-}\|\|y - y^{-}\| \tag{5}$$

Moreover, we select the top K sequences with the highest cross-correlation scores. Because the size of the collected QoS dataset is usually very large, it is very inefficient to perform a linear scan across the whole dataset. On the other hand, the geo-location of both users and services can be easily obtained by mapping the IP addresses to the geographical points using the pre collected database [17]. Because geographically closed user-service pairs have more chances to share the same IT infrastructures, such as routers and network workloads, they may have quite similar QoS values or trends, especially in terms of the sudden changes occurring over the time domain. Thus, the temporal QoS values of spatially close user-service pairs are very likely correlated. Here the spatial information from user-service pairs to effectively reduce the searching range while improving prediction performance has been employed.

We first testify our assumption that the temporal QoS values of spatially close user service pairs are very likely correlated. Large number of test cases from a QoS repository has been used. In each test case, we randomly choose a temporal QoS sequence ya corresponding to a user-service pair Pa and search for its most correlated temporal QoS sequence xb corresponding to user-service pair Pb within the different spatial similarities between Pa and Pb in all the other sequences.

Each validation result is the average of over 1,000 different test cases. We employ the spatial similarity simS to represent the spatial distance between two user-service pairs. Let Pa = (ua , va) and Pb = (ub , vb), where u and v denote the corresponding user and service. Spatial similarity simS between two user-service pairs means the average value of two geodesic distances between the two users and two services: simS(Pa , Pb) = 1/2 * (dist(ua , ub) + dist(va , vb)), where dist() denotes a geodesic distance function. We find that the most correlated QoS sequences of ya are more likely to be discovered in the spatially similar user-service pairs for Pa . Therefore, we can conclude that the spatial information of users and services can be utilized for discovering the

most correlated temporal QoS sequences. Figure 2 presents an example of a discovered xb corresponding to the user-service pair Pb with a very high correlation coefficient compared to ya corresponding to the user-service pair Pa .

Based on this analysis, we take advantage of the spatial information of user u and service v to predict the current temporal QoS sequence y, as well as the spatial information of each user ui and service vi of the collected reference temporal QoS sequences zi ($1 \preceq i \preceq$ M, where M is the number of other collected temporal QoS sequences). This reduces the search space for discovering the most correlated temporal QoS sequences for y. It is shown in Figure 3.2.

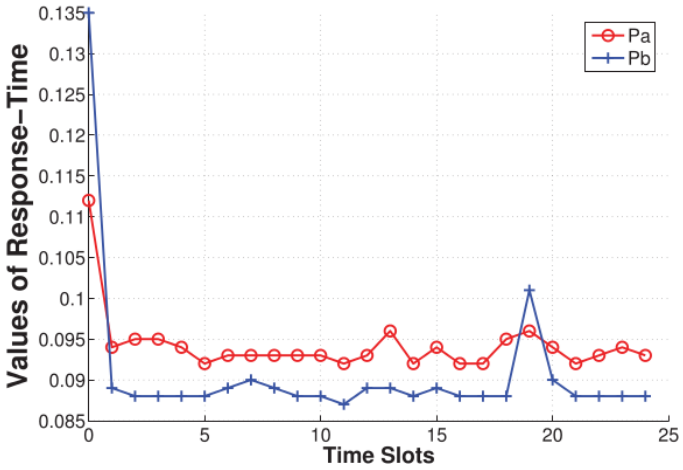


Figure 3.2: Temporal response time sequences: y a (corresponding to Pa) and xb (corresponding to Pb), S(ya , xb) = 0.955, simS(P a , P b) = 83.121km

Let P, which contains user u and service v, have a temporal sequence y. To find the most correlated temporal QoS sequences for y, we first retrieve a set of spatially similar user-service pairs GS(P). In order to efficiently obtain GS(P), we build a grid representation based on the geographical map and divide the map into multiple buckets, as shown in Figure 3. The length and width of each bucket are represented by differences in longitude and latitude, respectively. As in Wang et al. [2014], bucket length has been set to 0.1156 km and bucket width to 0.1491 km. Then, we map u, v, and each user ui and service vi of other temporal QoS sequences to a bucket, which are represented by u.bucket, v.bucket, ui.bucket, and vi.bucket, respectively. Note that we randomly select a set of user-service pairs from the repository when there is no user/service found in the same bucket. For each ui and vi , if u.bucket equals ui.bucket or v.bucket equals vi.bucket,

we calculate the spatial similarity between each selected Pi and P and select the top K user-service pair Pi as the discovered elements in GS(P). Furthermore, we calculate the correlation coefficient between y and each zi corresponding to the user-service pair Pi in GS(P) (Figure 3.3)and select the top K most correlated user-service pairs Pjs with the highest correlation coefficients as well as their corresponding temporal QoS sequence xj ($1 \preceq j \preceq$ K). Note that the correlation coefficients should be greater than zero.

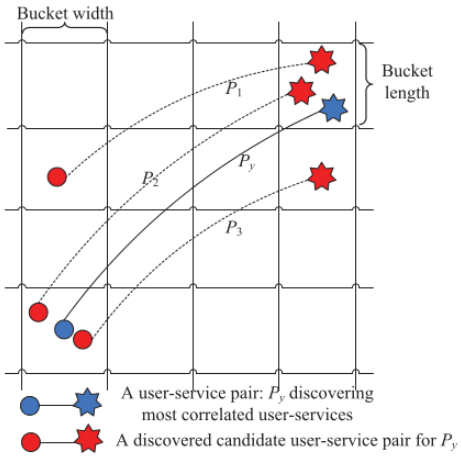


Figure 3.3: Discover GS(P) for P

4 Work Done

This section covers the work done for implementing spatial-temporal time-aware Quality of Service prediction framework.

4.1 *Experimental Framework*

The simulations were performed on a Linux system with 8 GB RAM, with CPU speed 2.60Hz and x64-based processor. The language of implementation used was Python.

4.2 *QoS prediction framework steps*

For each user, service pair in the dataset we obtain the temporal sequence for it from the dataset and store it as a dictionary in a file with (user, service) pair as the key and the temporal sequence to be the value.

We similarly retrieve the geographical information i.e latitude and longitude and store them as a dictionary with user ID as key and latitude and longitude as values and a similar process is performed for services and their geolocations.

We create a dictionary for mapping each user u and each service v to a bucket based on geolocation. The buckets are formed based on geolocations and the width and the length are set to 0.1491km and 0.1156km respectively. The buckets i and j where ui = u and vj = v are selected and the services belonging to this buckets are selected. The top k user, service pairs are selected based on geodesic distance. Out of these top k user-service pairs, k pairs are selected based on cross correlation scores.

We use the temporal sequences of top k user, service pairs to predict the QoS for the given user, service pairs at the nth time slot using Lasso regression.

We compute various scores for measuring the accuracy such as Mean Average error, root mean squared error etc.

4.3 *Dataset used*

We employ a real-world Web service QoS performance repository [Zhang et al. 2011] to evaluate the proposed approach. The repository contains a large number of temporal response time sequences collected from 142 distributed computers located in 57 countries from PlanetLab 1 to 4,532 distributed services all around the world. Each sequence

contains a set of temporal QoS values collected from a computer (user) for a service with at most 64 QoS values collected once after a time interval in a time slot. Each time slot lasts for 15 minutes, and the time interval between two adjacent time slots is 15 minutes. All the sequences are collected concurrently, lasting for 16 hours. Therefore, a 142 4,532 64 user-service-time matrix is constructed containing a particular response time value from the QoS invocation records in each of its positions. Some QoS values in the matrix are invalid; these are marked as zero. If the response time is larger than 20s, it is recorded as 20s in the datasets.

4.4 *Parameter Settings*

As discussed in the paper, the K value for the top correlated pairs to the current user-service pair was set to 10. The regularization coefficient was set to 0.1 as in the paper for Lasso regression.

4.5 *Results and Discussion*

The results for Lasso and Linear regressor for QoS prediction and the actual QOS value is shown in Figure 4.1 and the accuracy calculation is determined in Figure 4.2,4.3 and 4.4.

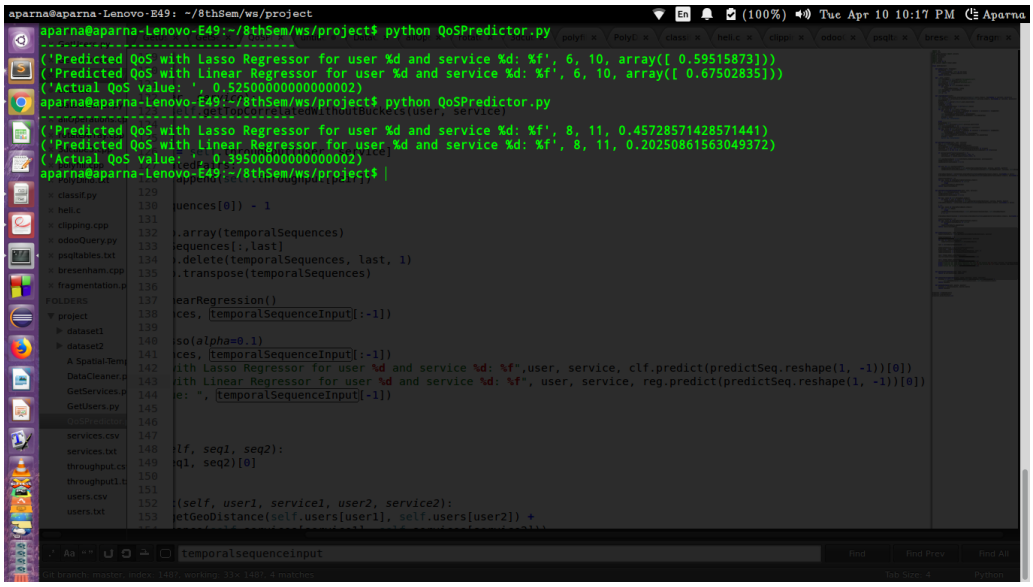


Figure 4.1: QoS Prediction

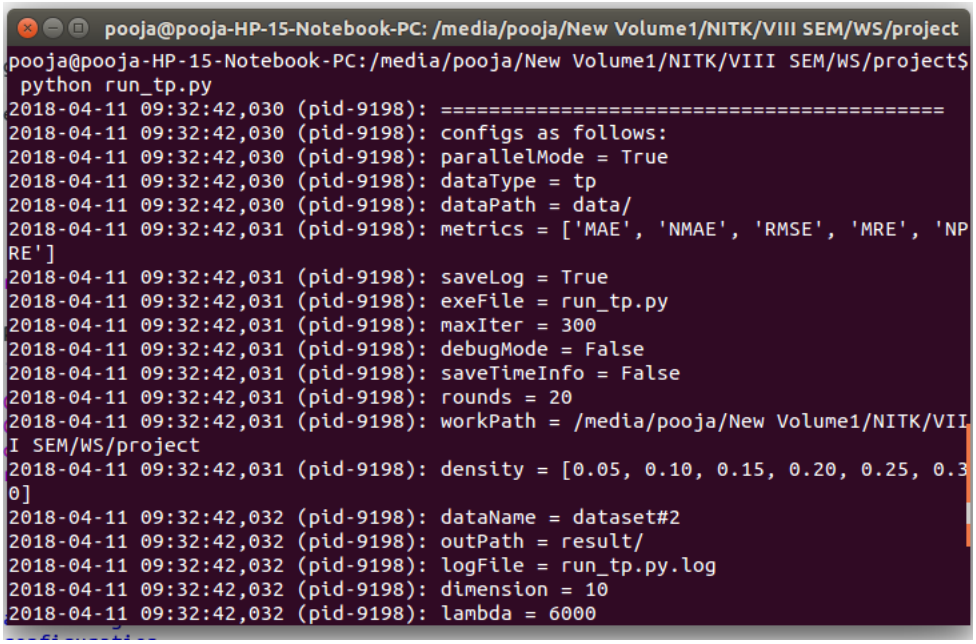


Figure 4.2: Accuracy Analysis

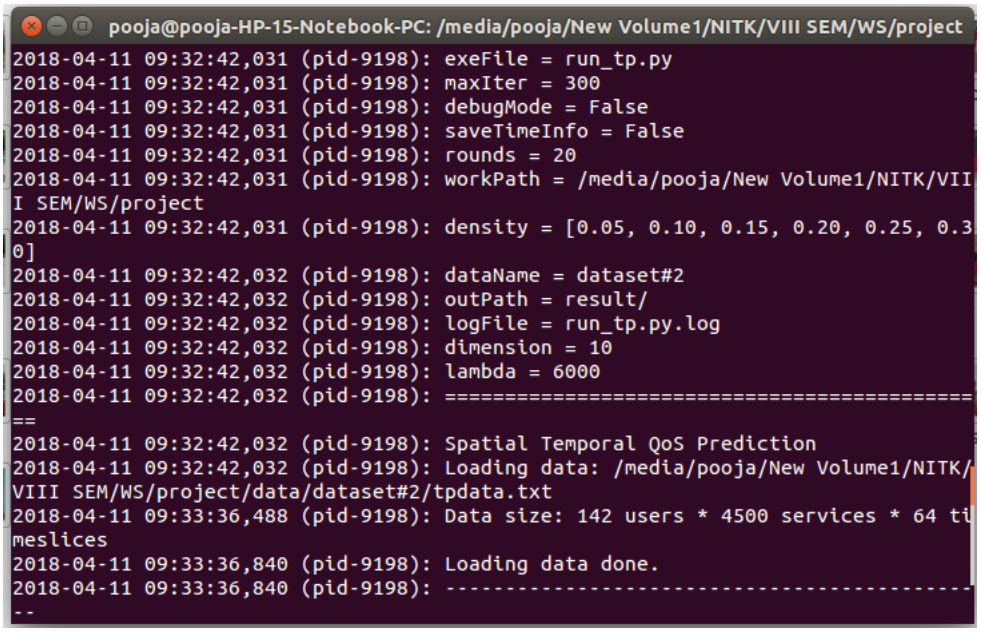
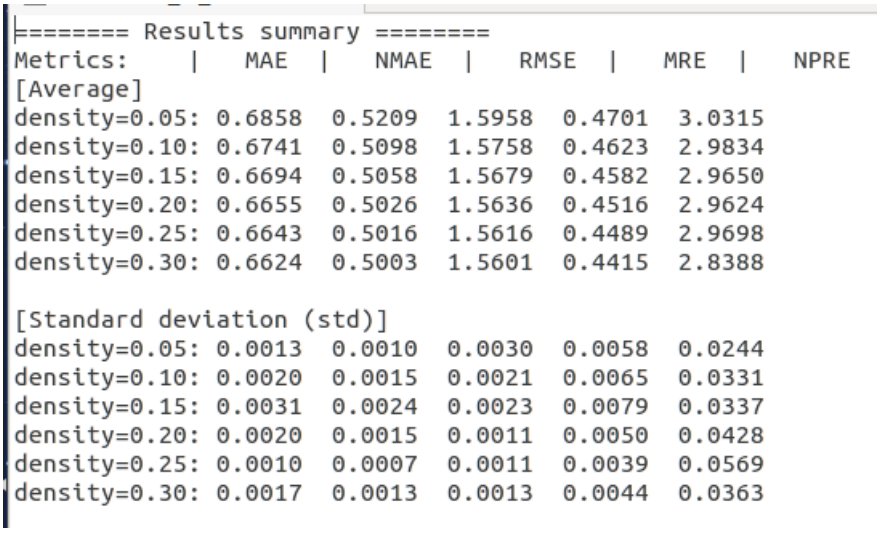


Figure 4.3: Accuracy Analysis part 2



```
======== Results summary ========
Metrics:    |   MAE  |   NMAE  |   RMSE  |   MRE  |   NPRE
[Average]
density=0.05: 0.6858  0.5209  1.5958  0.4701  3.0315
density=0.10: 0.6741  0.5098  1.5758  0.4623  2.9834
density=0.15: 0.6694  0.5058  1.5679  0.4582  2.9650
density=0.20: 0.6655  0.5026  1.5636  0.4516  2.9624
density=0.25: 0.6643  0.5016  1.5616  0.4489  2.9698
density=0.30: 0.6624  0.5003  1.5601  0.4415  2.8388

[Standard deviation (std)]
density=0.05: 0.0013  0.0010  0.0030  0.0058  0.0244
density=0.10: 0.0020  0.0015  0.0021  0.0065  0.0331
density=0.15: 0.0031  0.0024  0.0023  0.0079  0.0337
density=0.20: 0.0020  0.0015  0.0011  0.0050  0.0428
density=0.25: 0.0010  0.0007  0.0011  0.0039  0.0569
density=0.30: 0.0017  0.0013  0.0013  0.0044  0.0363
```

Figure 4.4: Accuracy and standard deviation

4.6 *Individual contribution of project members*

Aparna P L : Data processing, Grid formation and Spatial temporal technique implementation

Padmaja : Data collection and formatting and generic regressor

Pooja M S: Data pre-processing and lasso regressor

5 Conclusion and Future Work

In this project, a novel spatial temporal QoS prediction approach to time-aware Web service recommendation has been proposed. The temporal QoS prediction is formulated as a generic regression problem, where a zero-mean Laplace prior distribution assumption is made on the residuals of QoS prediction. Lasso regularization was introduced to facilitate the sparse representation of the temporal QoS sequence. Moreover, the geolocations of end users and services were employed to effectively retrieve the most similar QoS series. The extensive experimental results demonstrated that the proposed approach outperforms the state-of-the-art temporal QoS prediction methods for time- aware Web service recommendation.

The current method requires us to retrieve other QoS values at the current time slot, which cannot be applied to forecast future temporal QoS values. In the future, we will investigate an online algorithm to predict future QoS values based on accumulated data. Moreover, we will study the hierarchical indexing method to improve overall performance for the spatial-temporal QoS prediction.

5.1 *Complete work plan of the project*

Data collection and pre preprocessing- Jan 2018

Generic regressor - Feb 2018

Lasso regressor - Feb 2018

Spatial temporal grid formation - March 2018

Buckets, similarity calculation - March 2018

Accuracy measures - April 2018

References

[1] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng. Quality driven web services composition. In Proceedings of the 12th international conference on World Wide Web , pages 411421. ACM, 2003

[2] J. Zhang, J. Zhang, and H. Cai. 2007. Performance prediction based EX-QoS driven approach for adaptive service composition. Springer and Tsinghua University Press (2007)

[3] Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang. 2004. QoS aware middleware for web services composition. IEEE Transactions on Software Engineering 30, 5 (May 2004), 311327.

[4] A. Menasce. 2002. QoS issues in web services. IEEE Internet Computing 6, 6 (Nov Dec. 2002), 7275.

[5] Jaeger, G. Rojec-Goldmann, and G. Muhl. 2004. QoS aggregation for web service composition using workflow patterns. In Proceedings of the IEEE International Conference on Enterprise Computing. 149159.

[6] Godse, U. Bellur, and R. Sonar. 2010. Automating QoS based service selection. In Proceedings of the IEEE International Conference on Web Services. 510.

[7] Li, J. Huai, and H. Guo. 2009. An adaptive web services selection method based on the QoS prediction mechanism. In Proceedings of the IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technologies. 395402.

[8] Cavallo, M. D. Penta, and G. Canfora. 2010. An empirical comparison of methods to support QoS-aware service selection. In Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems. 6470.

[9] Zeng, C. Lingenfelder, H. Lei, and H. Chang. 2008. Event-driven quality of service prediction. In Proceedings of the 6th International Conference on Service-Oriented Computing. 147161

[10] Amin, A. Colman, and L. Grunske. 2012. An approach to forecasting QoS attributes of web services based on ARIMA and GARCH models. In Proceedings of the IEEE International Conference on Web Services. 7481.

[11] Hooman and P. J. Kennedy. 2009. HDAX: Historical symbolic modelling of delay time series in a communications network. In Proceedings of the 8th Australasian Data Mining Conference. 129137.

[12] Miller, I.Albert, S. K. Lam, J. A. Konstan, and J. Riedl. 2003. MovieLens unplugged: Experiences with an occasionally connected recommender system. In Proceedings of the ACM 2003 International Conference on Intelligent User Interfaces. 263266.

[13] Tang, Y. Jiang, J. Liu, and X. Liu. 2012. Location-aware collaborative filtering for QoS-Based service recommendation. In Proceedings of the IEEE International Conference on Web Services. 2429.

[14] Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun. 2013. Personalized qos-aware web service recommendation and visualization. IEEE Transactions on Service Computing 6, 1 (2013), 3547.

[15] Yao, Q. Sheng, A. Segev, and J. Yu. 2013. Recommending web services via combining collaborative filtering with content-based features. In Proceedings of the IEEE International Conference on Web Services. 4249.

[16] Luo, J. Yin, S. Deng, Y. Li, and Z. Wu. 2012. Collaborative web service QoS prediction with location-based regularization. In Proceedings of the IEEE International Conference on Web Services. 2429.

[17] Heath 2011 GeoLiteCity.dat.gz. Retrieved from http://www.maxmind.com/download/geoip/database/.