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

National Institute of Technology Karnataka

Web services mini project presentation

Under the guidance of
Ashwitha P
Assistant Lecturer
Department of Information Technology

Presented By
Padmaja B 14IT128
Aparna P L 14IT132
Pooja M S 14IT230

Introduction

- ❖ 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.

- ❖ Users extract a list of candidate Web services from different service brokers to compose complicated Service Oriented Applications (SOA), where one or more optimal services are selected to build the applications.

- ❖ Practically, the end users are from various geographical locations whose QoS values are greatly dependent on network conditions and network geographical locations.

Literature Survey

- ❖ Godse et al. [1] 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.
- ❖ Hooman and Kennedy [3] 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 memory-based CF methods employ user-rating data to compute the similarity between users or items and predict QoS values accordingly [McLaughlin and Herlocker [4]; Miller et al. [5]]

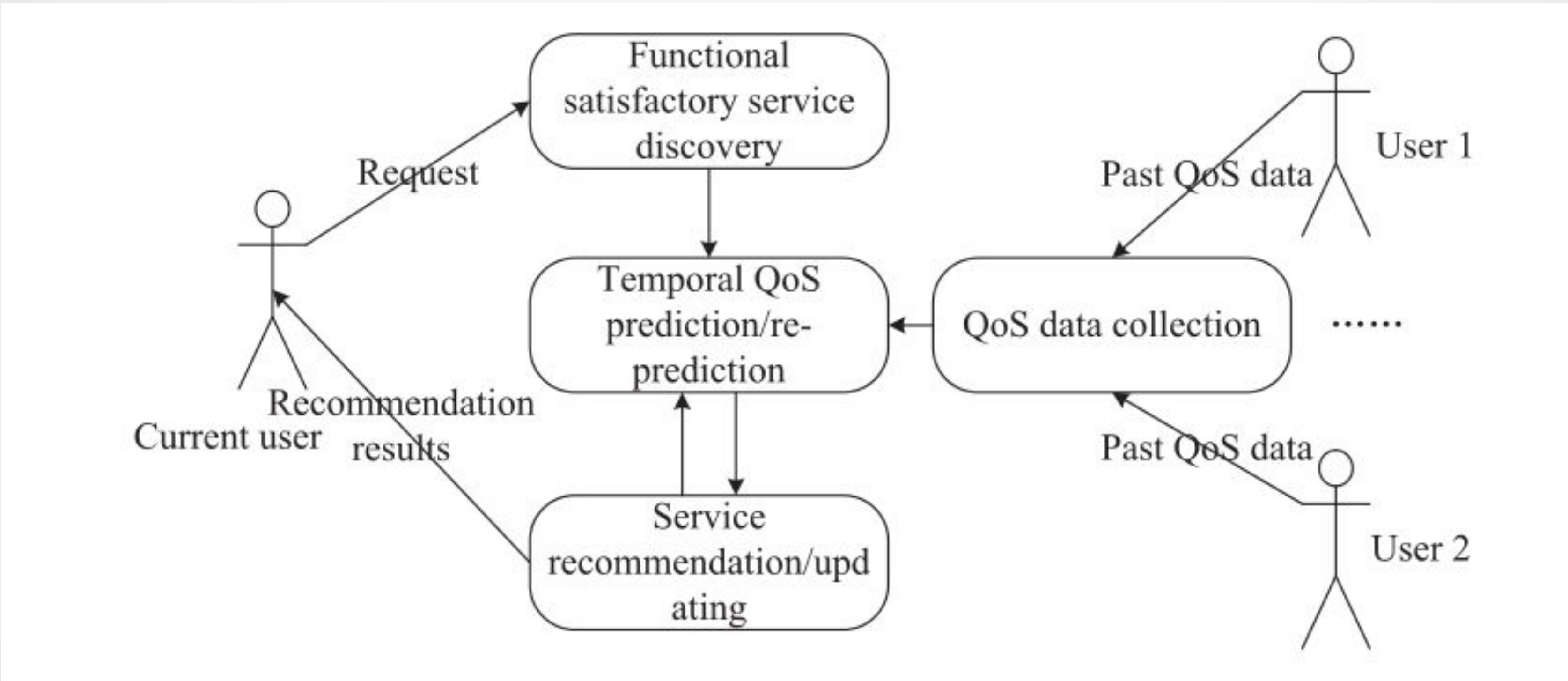
Problem Statement

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

Objectives

- Data processing for the grid and bucket formation steps
- Comparison between general regressor and Lasso regressor for time-aware prediction
- Using geolocation information like latitude, longitude for QoS prediction
- Selecting user-web service pairs based on cross-correlation and geodesic distance measure.

Research Methodology - System Architecture



Research Methodology - Generic regressor

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 $y = (y_1 , . . . , y_{n-1})^T$ denote the collected temporal QoS data for the user u and service v.

Research Methodology - Generic regressor

- QoS prediction can be solved using data driven methods, such as linear regression.
- Let w ∈ R^k*1 denote a vector of linear combination coefficients; then, the mapping function f (x) can be formulated as:
- $$f(x) = \sum_{i=1}^{K} w_i x_i = \mathbf{w}^\top \mathbf{x}.$$

- 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:

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{w}^\top \mathbf{X}\|_2^2 + \lambda_1 \|\mathbf{w}\|_2^2,$$

Research Methodology - Lasso Regressor

- We consider the zero mean Laplace distribution with probability density functions that have abrupt changes in gradient, which corresponds to a Lasso problem [Tibshirani][5]
- 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:

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{w}^{\top}\mathbf{X}\|_2^2 + \lambda\|\mathbf{w}\|_1,$$

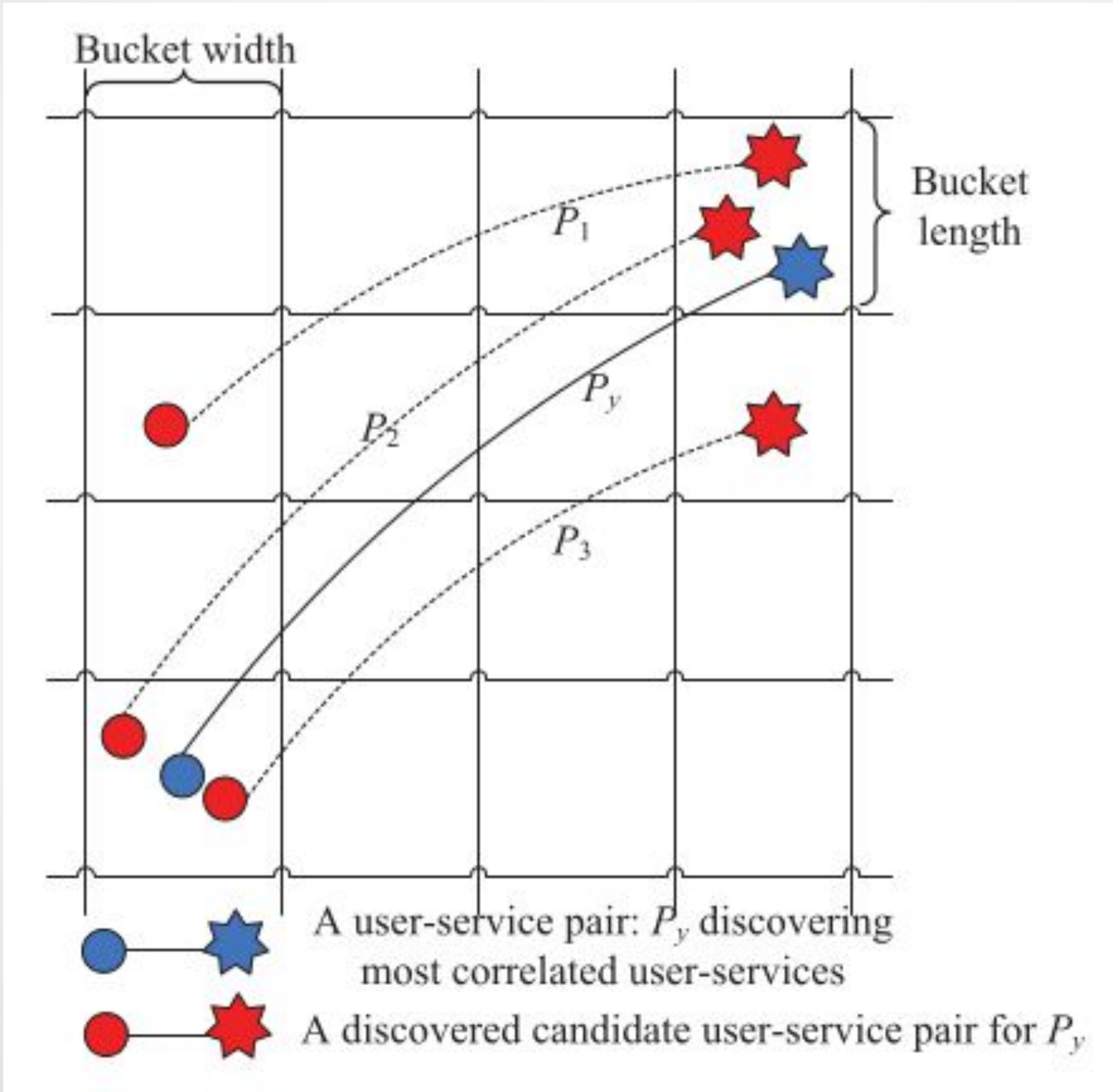
- Once the optimal linear combination coefficients w is obtained through Lasso, the QoS value for the current time slot can be directly predicted by mapping function f(x).

Research Methodology - 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:

$$S(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \bar{\mathbf{x}})^\top (\mathbf{y} - \bar{\mathbf{y}})}{\|\mathbf{x} - \bar{\mathbf{x}}\| \cdot \|\mathbf{y} - \bar{\mathbf{y}}\|}.$$

Research Methodology - Spatial-Temporal QoS Prediction --- Discover GSP(P)

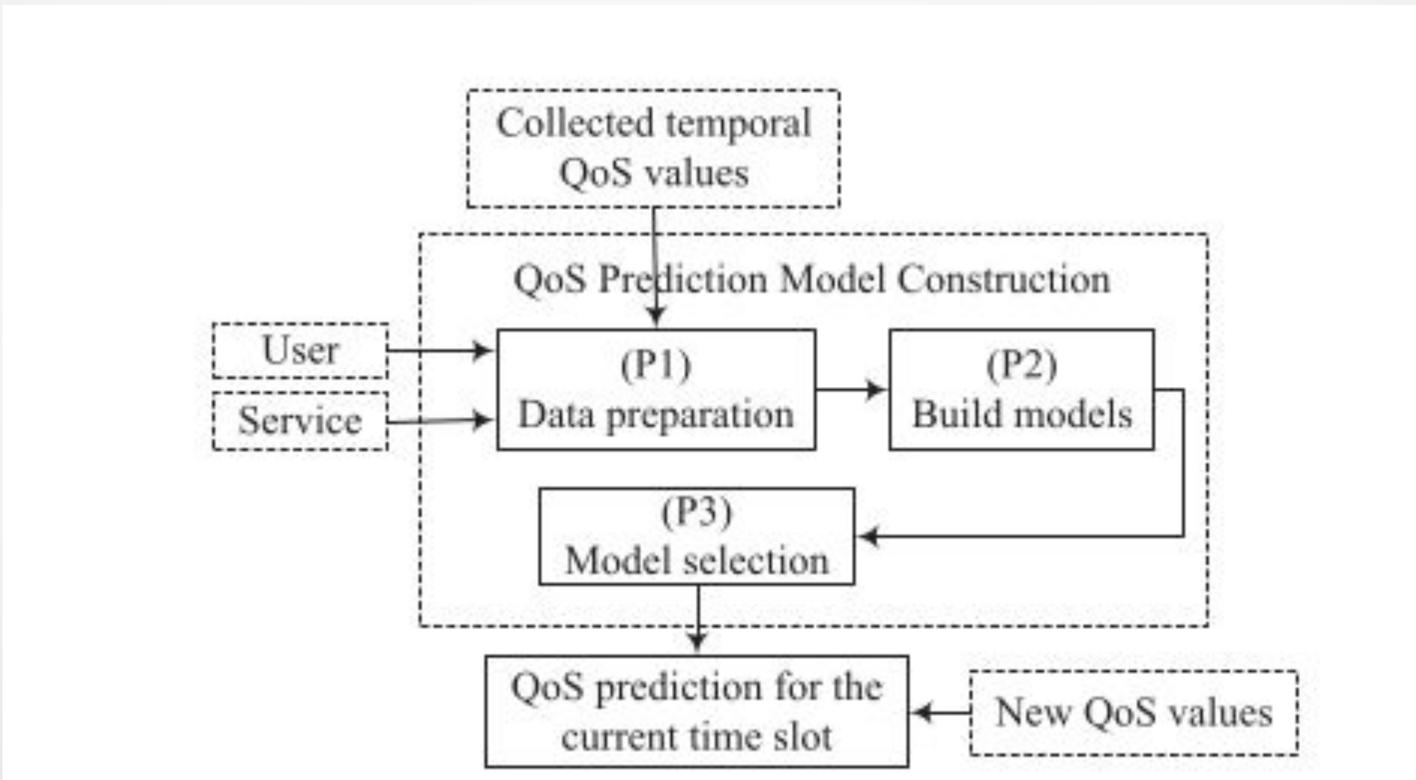


Work Done

- 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.

Work Done

Flow diagram for QoS prediction



Work Done

Description of 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 X 4,532 X 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.

Work Done

Parameter Setting:
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.

Results and Discussion

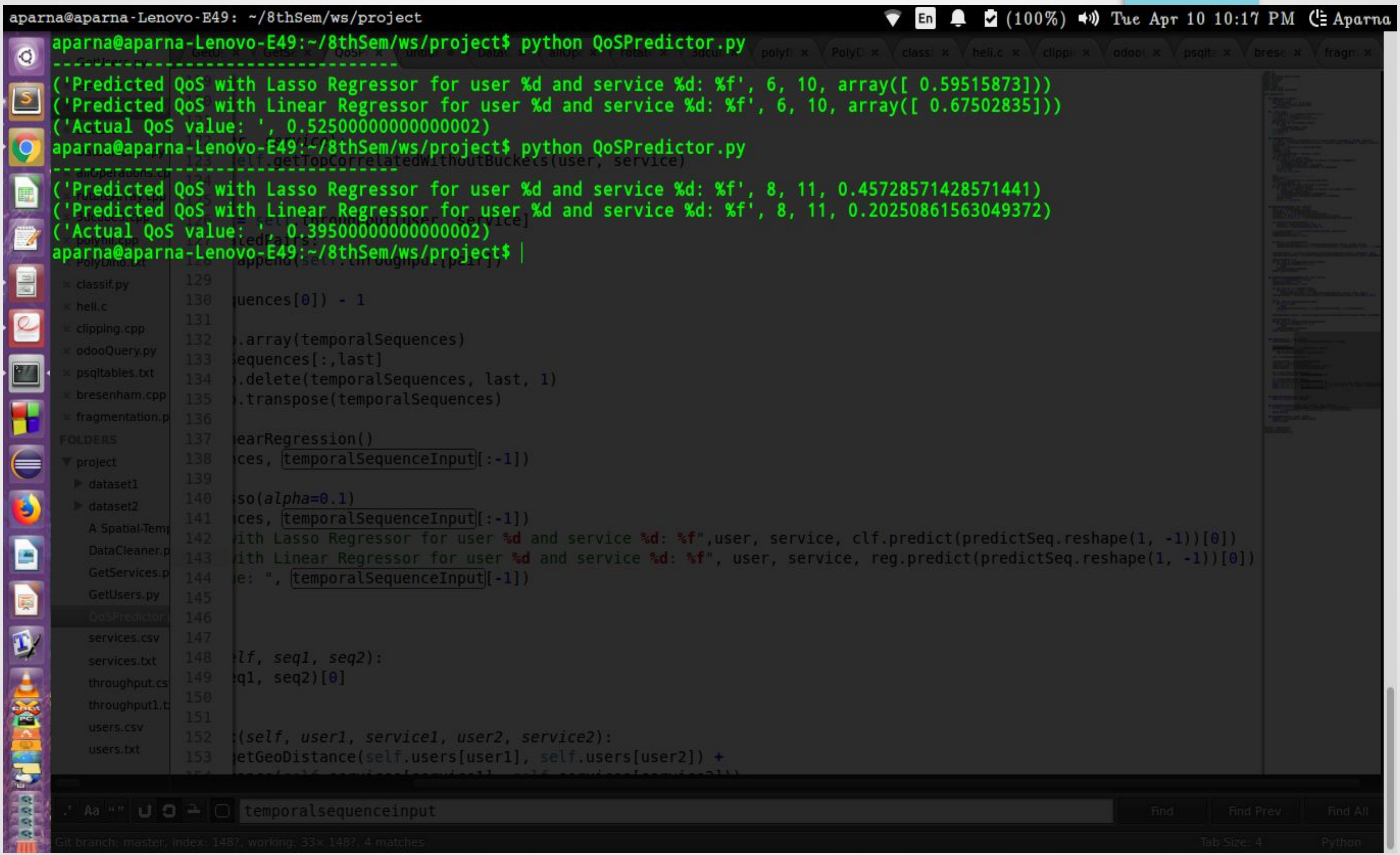
The following measures, (MAE, NMAE, RMSE) were used to measure accuracy of QoS prediction

$$MAE = \frac{\sum_{i,n} |Q(c_i, s_n) - \widehat{Q}(c_i, s_n)|}{M}$$

$$NMAE = \frac{MAE}{(\sum_{i,n} Q(c_i, s_n))/M}$$

$$RMSE = \sqrt{\frac{\sum_i (\hat{y}_n - y_n)^2}{N}},$$

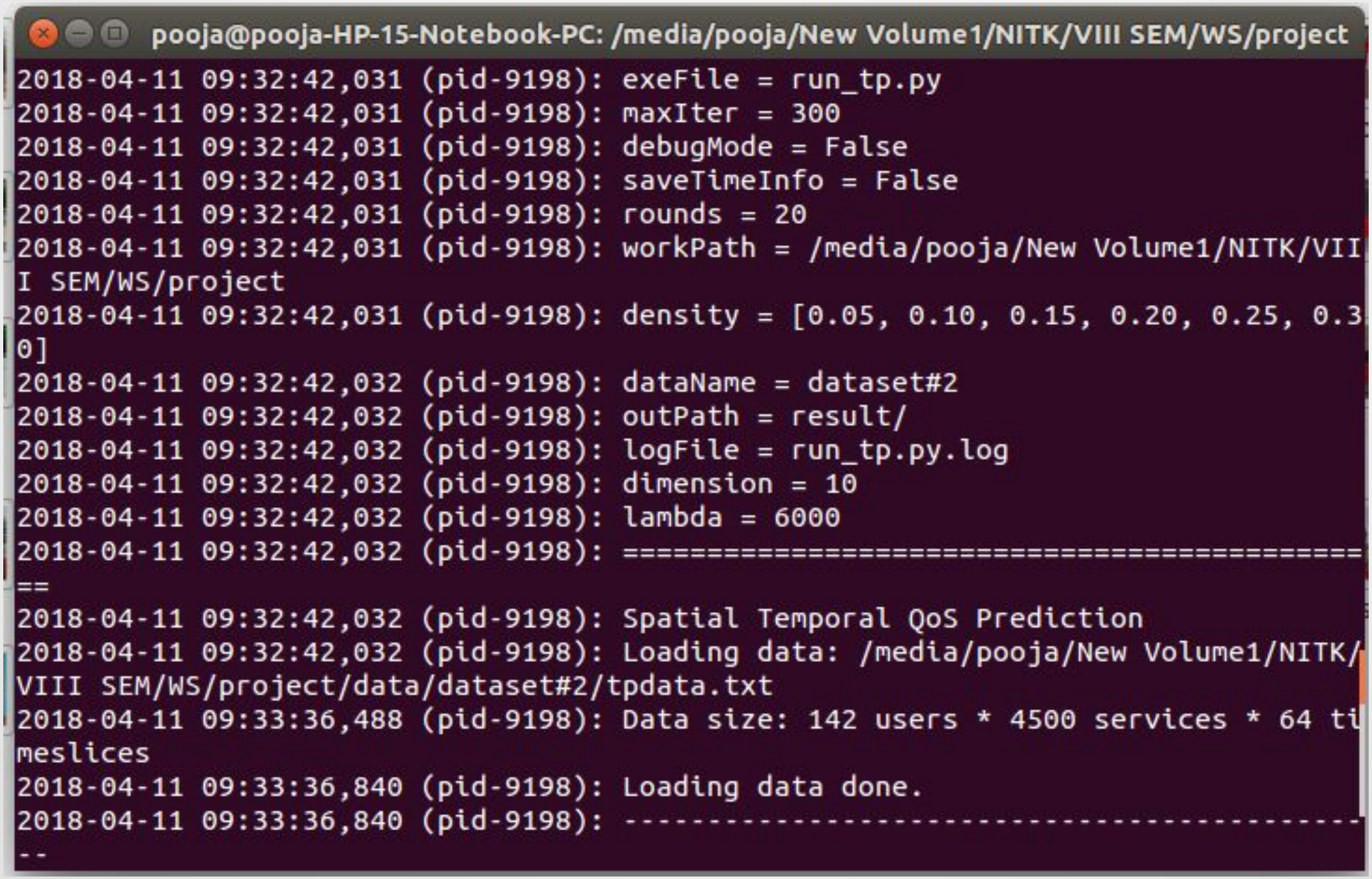
Results and Discussion



Results and Discussion

```
pooja@pooja-HP-15-Notebook-PC: /media/pooja/New Volume1/NITK/VIII SEM/WS/project
pooja@pooja-HP-15-Notebook-PC:/media/pooja/New Volume1/NITK/VIII SEM/WS/project$
 python run_tp.py
2018-04-11 09:32:42,030 (pid-9198): ==========================================
2018-04-11 09:32:42,030 (pid-9198): configs as follows:
2018-04-11 09:32:42,030 (pid-9198): parallelMode = True
2018-04-11 09:32:42,030 (pid-9198): dataType = tp
2018-04-11 09:32:42,030 (pid-9198): dataPath = data/
2018-04-11 09:32:42,031 (pid-9198): metrics = ['MAE', 'NMAE', 'RMSE', 'MRE', 'NP
RE']
2018-04-11 09:32:42,031 (pid-9198): saveLog = True
2018-04-11 09:32:42,031 (pid-9198): exeFile = run_tp.py
2018-04-11 09:32:42,031 (pid-9198): maxIter = 300
2018-04-11 09:32:42,031 (pid-9198): debugMode = False
2018-04-11 09:32:42,031 (pid-9198): saveTimeInfo = False
2018-04-11 09:32:42,031 (pid-9198): rounds = 20
2018-04-11 09:32:42,031 (pid-9198): workPath = /media/pooja/New Volume1/NITK/VII
I SEM/WS/project
2018-04-11 09:32:42,031 (pid-9198): density = [0.05, 0.10, 0.15, 0.20, 0.25, 0.3
0]
2018-04-11 09:32:42,032 (pid-9198): dataName = dataset#2
2018-04-11 09:32:42,032 (pid-9198): outPath = result/
2018-04-11 09:32:42,032 (pid-9198): logFile = run_tp.py.log
2018-04-11 09:32:42,032 (pid-9198): dimension = 10
2018-04-11 09:32:42,032 (pid-9198): lambda = 6000
```

Results and Discussion



pooja@pooja-HP-15-Notebook-PC: /media/pooja/New Volume1/NITK/VIII SEM/WS/project

```
2018-04-11 09:32:42,031 (pid-9198): exeFile = run_tp.py
2018-04-11 09:32:42,031 (pid-9198): maxIter = 300
2018-04-11 09:32:42,031 (pid-9198): debugMode = False
2018-04-11 09:32:42,031 (pid-9198): saveTimeInfo = False
2018-04-11 09:32:42,031 (pid-9198): rounds = 20
2018-04-11 09:32:42,031 (pid-9198): workPath = /media/pooja/New Volume1/NITK/VII
I SEM/WS/project
2018-04-11 09:32:42,031 (pid-9198): density = [0.05, 0.10, 0.15, 0.20, 0.25, 0.3
0]
2018-04-11 09:32:42,032 (pid-9198): dataName = dataset#2
2018-04-11 09:32:42,032 (pid-9198): outPath = result/
2018-04-11 09:32:42,032 (pid-9198): logFile = run_tp.py.log
2018-04-11 09:32:42,032 (pid-9198): dimension = 10
2018-04-11 09:32:42,032 (pid-9198): lambda = 6000
2018-04-11 09:32:42,032 (pid-9198): ============================================
==
2018-04-11 09:32:42,032 (pid-9198): Spatial Temporal QoS Prediction
2018-04-11 09:32:42,032 (pid-9198): Loading data: /media/pooja/New Volume1/NITK/
VIII SEM/WS/project/data/dataset#2/tpdata.txt
2018-04-11 09:33:36,488 (pid-9198): Data size: 142 users * 4500 services * 64 ti
meslices
2018-04-11 09:33:36,840 (pid-9198): Loading data done.
2018-04-11 09:33:36,840 (pid-9198): --------------------------------------------
--
```

Results and Discussion

```
======== Results summary ========
Metrics:     |    MAE   |    NMAE   |    RMSE
[Average]
density=0.05: 0.6858  0.5209  1.5958  0.
density=0.10: 0.6741  0.5098  1.5758  0.
density=0.15: 0.6694  0.5058  1.5679  0.
density=0.20: 0.6655  0.5026  1.5636  0.
density=0.25: 0.6643  0.5016  1.5616  0.
density=0.30: 0.6624  0.5003  1.5601  0.

[Standard deviation (std)]
density=0.05: 0.0013  0.0010  0.0030  0.
density=0.10: 0.0020  0.0015  0.0021  0.
density=0.15: 0.0031  0.0024  0.0023  0.
density=0.20: 0.0020  0.0015  0.0011  0.
density=0.25: 0.0010  0.0007  0.0011  0.
density=0.30: 0.0017  0.0013  0.0013  0.
```

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.

Individual Contribution

- ❖ 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 QoS prediction

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 411–421. 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), 311–327.

[4] A. Menasce. 2002. QoS issues in web services. IEEE Internet Computing 6, 6 (Nov – Dec. 2002), 72–75.

[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. 149–159.

[6] Godse, U. Bellur, and R. Sonar. 2010. Automating QoS based service selection. In Proceedings of the IEEE International Conference on Web Services. 5–10.