Road Travel Time Prediction using Vehicular Network

Sejoon Lim, *Member, IEEE*

***Abstract*— In this paper, we compare the road travel time prediction performance of the regularized least squares (RLS) and sparsity-based regularization (SBR) methods. Using real travel time data gathered by probe vehicles connected to a GPS sensor network, we demonstrate that SBR methods show a better prediction performance than simple RLS algorithms. We further suggest that feature mapping be applied in the SBR algorithm, which results in even better prediction performance.**

***Index Terms*— Vehicular network, Traffic estimation, GPS, Sparsity-based Regularization.**

I. Introduction

ACCURATE traffic prediction is important for intelligent transport systems. Previous studies focused on highway traffic prediction where traffic data are available [1]–[6]. However, this type of research is limited by the amount of accessible data. Since most traffic monitoring is performed by using sensors installed on the roads and there are not many sensors, the available data are limited to those for a small set of road segments.

In this study, large-scale data gathered by probe vehicles equipped with GPS sensor nodes were used. The probe vehicles—more than 20 taxis operating in the Boston area—reported their location and time information almost every second while they were traveling. We used this information and underlying map information to determine the length of time it took a car to travel along specific road segments, which yielded a large amount of travel time information for a large set of road segments in the Boston area. The scope of the information was unlimited, since the probe vehicles could travel wherever roads exist, and the GPS sensor nodes recorded their trajectory for every journey. Furthermore, the dataset thus acquired can be expected to grow as the number of probe vehicles increases. Thus, a prediction model that can handle a large amount of data is required. In this paper, we therefore propose a prediction method that can utilize a large dimension of data.

In particular, the goal of this study was to predict the travel time for a road segment using the historical and current information of the given road segment, as well as of other spatially related road segments. The input vector of the learning system was historical and current travel time information for spatially related road segments and the output was the travel time for the given road segment.

One of the most general approaches for time series prediction is the regularized least squares (RLS) method [7], [8]. However, the RLS approaches that are frequently used can result in over fitting when the number of samples is smaller than the number of features. Since the travel conditions of a road segment are considered to depend on those for a subset of all the road segments, sparsity-based regularization (SBR) is a good approach for addressing this problem [9]–[12]. SBR benefits in terms of prediction performance from the fact that it shrinks the value of some features to 0 rather than retaining their lower values. It also benefits in terms of computational performance from the fact that it retains a small set of features when the original input space domain is very large.

We used SBR algorithms for prediction and compared their performance with that of RLS methods. We further extended the input space to incorporate a higher order relationship between the elements of the input space.

II. GPS Data and a Map

A. GPS data

In [13], a vehicular sensor network system called CarTel was developed that uses GPS and wireless communication to collect position and time data from cars. In this study, CarTel nodes were deployed in more than 20 taxis operating in the Boston area and used to collect travel data. These data were organized per road segment to create a historical database of traffic delays. The parameters of the data points acquired from a sensor node consisted of latitude, longitude, and the time stamp. We acquired a large amount of this kind of data point collected by the taxis' CarTel nodes.

B. Data organized on a map

The taxis' GPS trajectories were matched to the corresponding road segments on the underlying map obtained from Navteq [14]. By matching GPS samples with points on a map, we could determine the amount of delay for each road segment. We acquired about 6 million such delay samples spanning a year. Although Navteq data (Fig. 1) include information for about 600,000 road segments, we combined them by using statistical similarities to generate only 40,000 groups of road segments, which is a sufficiently large amount to retain the meaningfulness of the delay data.

Manuscript received July 21, 2015. Sejoon Lim is with the Kookmin University, Seoul, South Korea (phone: +82-2-910-5469; e-mail: lim@kookmin.ac.kr).

Fig. 1. Navteq data for road segments in Boston.

III. Problem Setting

A. Objective

The goal of this study was to predict the road travel time for a road based on the knowledge about the road itself, as well as about other roads. In particular, the travel time of the road indicated as Road 0 in Fig. 2 was predicted based on the travel time information of the roads indicated as Roads 0 to 5. While many prediction problems assume that traffic data can be acquired at regular intervals, this assumption does not hold for our data gathering system, since the taxis with GPS sensor nodes traveled randomly according to the directions they received. Thus, we could not expect to acquire travel time data samples for a certain road segment at regular intervals. To simplify the prediction problem, we used the most recent N samples from a certain time period, regardless of the frequency with which the information was observed in a given time interval.

B. Input and output

The input and output pairs are described as follows.

Input *x*: a vector with the following elements:

- Recent N time and delay pairs of the road in question
- Recent M time and delay pairs of the other R roads
- Recent L time and delay pairs of the other R roads for the same hour of day
- The time of the observed delay.

Output *y*: a scalar representing the observed delay for the road in question.

Here, "time" is defined as the seconds from the start of the day and "delay" as the travel time in seconds. The dimension p of the input vector x is $p = 2N + 2MR + 2LR + 1$. For example, when $N = 10, M = 10, L = 10$, and $R = 10$, the dimension is 421. This is relatively large considering that only 5 to 20 samples per day were acquired. The dimension of the feature space may become even larger when a feature map that maps an input space into a larger feature space is used.

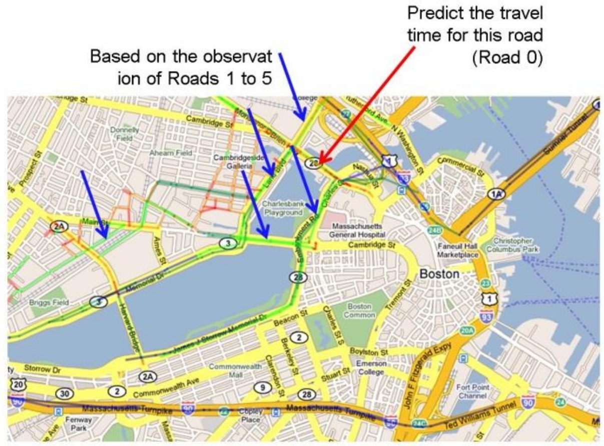

Fig. 2. Road 0 is the road for which the travel time was predicted using the travel time data collected for Roads 0-5.

IV. Learning Methods

To learn the function *f* that maps input *x* to output *y*, RLS, SBR, and their modifications were used.

A. Regularized least squares (RLS)

Tikhonov regularization with the least square loss function minimizes the expression

$$\frac{1}{n}\sum_{i=1}^{n} |y_i - f(x_i)|^2 + \lambda \|f\|_H^{\,2}, \quad (1)$$

Where $f(x_i)$ is the estimate of y_i, $\|\cdot\|_H$ is the norm in the function space H. Suppose that we have a kernel function $k(x_i, x_j)$ as follows:

Linear :	$k(x_i, x_j) = x_i^t x_j$
Quadratic:	$k(x_i, x_j) = (x_i^t x_j + 1)^2$
Gaussian :	$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$

Accordingly, a minimizing function f can be expressed as

$$f(x) = \sum_{i=1}^{n} c_i k(x_i, x), \quad (2)$$

for some *n*-tuple $(c_1, c_2, \ldots, c_n)$.

Thus, the RLS problem can be solved in terms of a kernel as

$$c^* = argmin_{c \in R^n} \frac{1}{2}\|y - Kc\|_2^{\,2} + \frac{\lambda}{2}\|f\|_K^{\,2}, \quad (3)$$

$$c^* = (K + \lambda I)^{-1} y, \quad (4)$$

where K is the kernel matrix with each element satisfying $K_{ij} = k(x_i, x_j)$, and $\|\cdot\|_K$ is the reproducing kernel Hilbert spaces (RKHS) norm[15]. λ can be selected to minimize the generalization errors.

B. Sparsity-based regularization (SBR)

A sparse solution can be found by using L_0 norm instead of L_2 norm in Tikhonov regularization. However, L_1 norm is frequently used as an approximation of L_0 norm, since it is difficult to manage L_0 norm. Our SBR problem can be expressed using L_1 norm as

$$\frac{1}{n}\sum_{i=1}^{n} |y_i - \beta x_i|^2 + \lambda \|\beta\|_1 \quad (5)$$

A popular method for solving the above minimization problem is the iterative thresholding algorithm [9], [16]. Let β^λ be a solution of the convex relaxation problem; then, it can be proved that the following iterative algorithm converges to β^λ as the number of iteration increases:

$$\beta_0^\lambda = 0,\ \beta_t^\lambda = \mathrm{S}_\lambda[\beta_{t-1}^\lambda + \tau X^T(\mathrm{Y} - \mathrm{X}\beta_{t-1}^\lambda)], \quad (6)$$

where

$$\mathrm{S}_\lambda(\beta_i) = \begin{cases} \beta_i + \frac{\lambda}{2}, & if\ \ \beta_i < -\frac{\lambda}{2} \\ 0, & if\ \ |\beta_i| \le -\frac{\lambda}{2} \\ \beta_i - \frac{\lambda}{2}, & if\ \ \beta_i > \frac{\lambda}{2} \end{cases} \quad (7)$$

λ and τ can be selected to minimize generalization errors.

C. SBR with feature map

We introduce a feature map Φ that maps an input space to a feature space.

$$\frac{1}{n}\sum_{i=1}^{n} |y_i - \beta\Phi(x_i)|^2 + \lambda \|\beta\|_1 \quad (8)$$

Two feature maps are described in the following and their prediction performance is compared in Section V.

1) Second order feature map: The following feature map incorporates the second order relationship between input features. In addition to the original input space, second order terms for every pair of input elements are added.

$$\Phi(x_1, x_2, \dots, x_p) = (x_1, x_2, \dots, x_p, x_1^2, x_1x_2, \dots, x_1x_p, \dots, x_px_1, \dots, x_p^2) \quad (9)$$

As shown in Section V, using this feature map, SBR results in a prediction performance that is only slightly improved or is even worse than that using the original input space. Moreover, it is also computationally expensive because the dimension of the feature space is large.

2) Selected higher order feature map: In light of the problems related to the second order feature map, we suggest a feature map that has second and third order terms for only related (time, delay) pairs. The intuition behind this selection of features is that the (time, delay) pairs comprise meaningful information only when they are considered together, while other random combinations of two input elements do not generate meaningful features.

$$\Phi(x_1, x_2, \dots, x_p) = (x_1, x_2, \dots, x_p, x_i^2, x_ix_j, x_j^2, x_i^3, x_i^2x_j, x_ix_j^2, x_j^3) \quad (10)$$

where (i, j) denotes every (time, delay) pair.

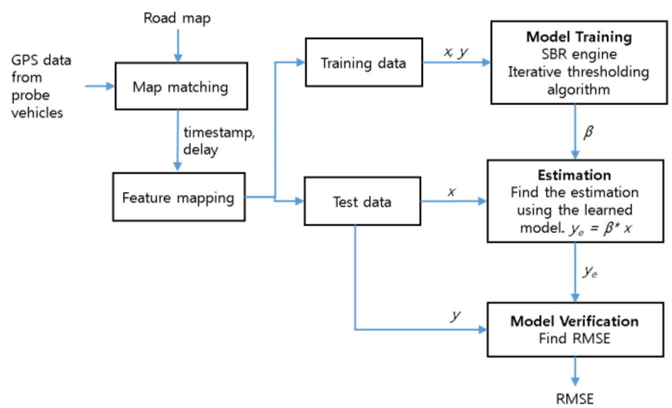

Fig. 3. Block diagram of the road travel time prediction system using sparsity-based regularization

V. RESULTS

We implemented the above algorithms in MATLAB. Table 1 shows the generalization errors for each method. Travel time data of 20 days were used to train the predictor model with parameters N = M = L = R = 10. To test the prediction quality, the root mean square (RMS) error was computed using the travel time data of five days.

As shown in Table 1, the SBR performs better than the RLS method. The effect of spatial information is noteworthy and affects RLS and SBR differently. A comparison of the two rows "False" and "True," i.e., those presenting the results when other roads' information was not or was used, respectively, reveals that, whereas in RLS the prediction performance is not improved by using other road's information, in SBR it is. The intuition behind this is that simple RLS algorithms tend to over fit the training data and do not generalize well, in particular, when the input space contains information unrelated to the prediction, whereas, in the case of SBR, the unnecessary information tends to be removed effectively by selecting only related features.

Figs. 4 and 5 show the detailed plots of the prediction. We can also observe that the SBR method using the selected second and third order feature map performs better than the other two SBR methods. This SBR performs better than that using the original input space because the higher order terms help to capture the pair-wise related information better.

TABLE I
GENERALIZATION ERROR COMPARISON

Use other roads	RLS			SBR		
	Linear	Quadratic	Gaussian	Basic	2nd order	Selected higher order
False	49.82	51.27	51.18	49.34	49.20	49.12
True	73.12	67.62	52.68	48.39	58.65	44.64

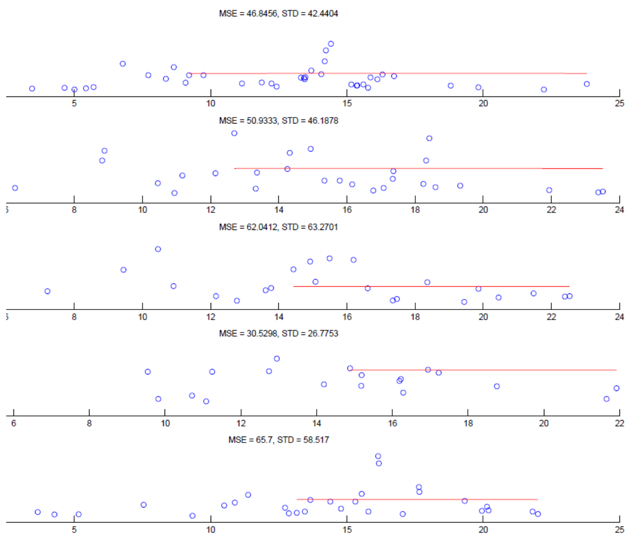

Fig. 4. Time series of travel time observation (circles) and the prediction using Gaussian RLS (line)

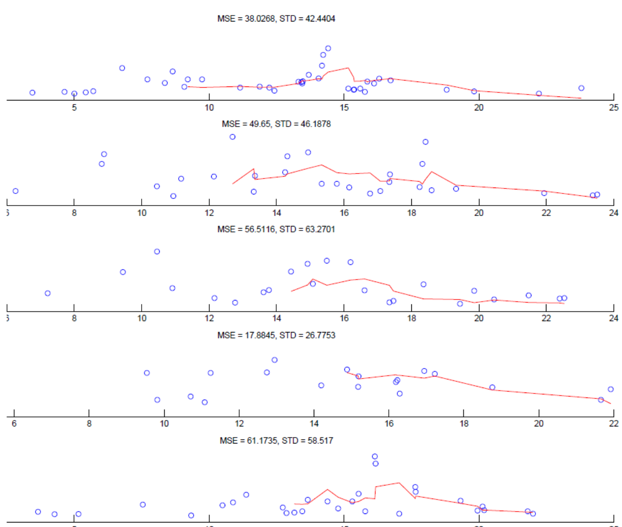

Fig. 5. Time series of travel time observation (circles) and the prediction using SBR with selected higher order feature map (line)

Moreover, by comparing the performance of this selection of features and the complete second-order feature map, we can conclude that using prior knowledge of the relationship between each input element facilitates prediction without leading to the over fitting that can occur if an excessive number of features are used.

VI. CONCLUSION

The performance of the RLS and SBR methods was investigated for road traffic prediction using real travel time data acquired from a GPS sensor network. SBR methods showed a superior performance, in particular, when the size of the input vectors was large as compared to the available sample number. The improved prediction was achieved by using a feature map that takes the relationship between the input spaces into account.

REFERENCES

[1] W.-H. Lin. “A gaussian maximum likelihood formulation for short term forecasting of traffic flow,” *IEEE Conference on Intelligent Transportation Systems*, Oakland, CA, USA, August, 2001.
[2] C.-H. Wu, J.-M. Ho, and D. T. Lee. “Travel-time prediction with support vector regression,” *IEEE Trans. ON Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276-281, 2004.
[3] H. Xu and D. J. Dailey. “Real time highway traffic simulation and prediction using inductance loop data,” *In Vehicle Navigation and Information Systems Conference*, Seattle, WA, USA, July 1995.
[4] Zhao, Jiandong, et al. “Travel time prediction based on pattern matching method, *“International Journal on Smart Sensing & Intelligent Systems*, vol. 8, no. 1, 2015.
[5] Oh, Simon, et al. “Short-term Travel-time Prediction on Highway: A Review of the Data-driven Approach.” *Transport Reviews* vol. 35, no. 1, pp. 4-32, 2015.
[6] Rasyidi, Mohammad Arif, and Kwang Ryel Ryu. “Short-Term Speed Prediction on Urban Highways by Ensemble Learning with Feature Subset Selection, *“Database Systems for Advanced Applications.* Springer Berlin Heidelberg, 2014.
[7] Rifkin, R., Yeo, G., & Poggio, T. “Regularized least-squares classification,” *Nato Science Series Sub Series III Computer and Systems Sciences*, vol. 190, pp. 131-154, 2003.
[8] De Vito, E., Caponnetto, A., & Rosasco, L. “Model selection for regularized least-squares algorithm in learning theory,” *Foundations of Computational Mathematics*, vol. 5, no. 1, pp. 59-85, Chicago, 2005.
[9] Daubechies, I., Defrise, M., and De Mol, C. “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on Pure and Applied Mathematics*, vol. 57, pp. 1413–1457, 2004.
[10] Rosasco, L., Villa, S., Mosci, S., Santoro, M. and Verri, “A. Nonparametric sparsity and regularization,” *J. Mach. Learn. Res*, vol. 14 pp. 1665–1714, 2013.
[11] S. Mosci, L. Rosasco, M. Santoro, A. Verri and S. Villa, Solving “Structured Sparsity Regularization with Proximal Methods,” in *Machine Learning and Knowledge discovery in Databases European Conference*, pp. 418-433, 2010.
[12] Hlavácková-Schindler, Katerina, Valeriya Naumova, and Sergiy Pereverzyev Jr. “Multi-penalty regularization for detecting relevant variables,” *Applied Mathematics*, 2014.
[13] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, and S. Madden. “CarTel: A Distributed Mobile Sensor Computing System,” *In 4th ACM SenSys*, Boulder, CO, November, 2006.
[14] Navteq. http://www.navteq.com/developer/index.htm
[15] Berlinet, Alain, and Christine Thomas-Agnan. “Reproducing kernel Hilbert spaces in probability and statistics, ”*Springer Science & Business Media*, 2011.
[16] Fornasier, Massimo, and Holger Rauhut. “Iterative thresholding algorithms,” *Applied and Computational Harmonic Analysis* vol. 25, no. 2, pp. 187-208, 2008.

Sejoon Lim received his Ph.D. degree in the department of Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, USA, in 2012. He is currently an Assistant Professor at Kookmin University, Seoul, South Korea. He was a senior member of technical staff at Oracle Corporation, Redwood City, CA, USA before joining Kookmin University. His research interests lie in traffic data analysis, traffic modeling and estimation, vehicle routing under uncertainty, multi-agent routing algorithm, and intelligent transportation systems.