2014 IEEE International Conference on Data Mining

Robust Dynamic Trajectory Regression on Road Networks: A Multi-Task Learning Framework

Aiqing Huang, Linli Xu*, Yitan Li and Enhong Chen
School of Computer Science and Technology
University of Science and Technology of China, Hefei, Anhui, China
cassie89@mail.ustc.edu.cn, linlixu@ustc.edu.cn, etali@mail.ustc.edu.cn, cheneh@ustc.edu.cn*

***Abstract*—Trajectory regression, which aims to predict the travel time of arbitrary trajectories on road networks, attracts significant attention in various applications of traffic systems these years. In this paper, we tackle this problem with a multi-task learning (MTL) framework. To take the temporal nature of the problem into consideration, we divide the regression problem into a set of sub-tasks of distinct time periods, then the problem can be treated in a multi-task learning framework. Further, we propose a novel regularization term with which we exploit the block sparse structure to augment the robustness of the model. In addition, we incorporate the spatial smoothness over road links and thus achieve a spatial-temporal framework. An accelerated proximal algorithm is adopted to solve the convex but non-smooth problem, which will converge to the global optimum. Experiments on both synthetic and real data sets demonstrate the effectiveness of the proposed method.**

***Keywords*-trajectory regression; multi-task learning; dynamic; structured sparsity;**

I. Introduction

Recent advances in satellites, the Global Positioning System (GPS) and tracking facilities have made it possible to collect a great amount of traffic data, which allows us to track a vehicle's moving path with its consecutive locations (also known as a trajectory). In this context, new challenges are introduced to data mining communities and investigations including trajectory clustering [1], classification [2] and regression [3]–[5] have been conducted.

In general, trajectory analysis needs to leverage the latent costs of links on road networks, which are essential information for analyzing traffic data. However, traffic cost of each road segment cannot be obtained easily and reliably in reality, whereas the total travel time over a complete trajectory could be measured and recorded more directly [4].

The problem of "trajectory regression" is thus raised as a recipe to address the issues above, where one seeks to learn the latent costs of links given a set of trajectories and the corresponding total costs. It differs from the traditional regression problem in that the instances (trajectories) are extremely sparse due to the fact that a driving path always spans just a small fraction of road segments while the traffic history data in some regions is extremely insufficient. Moreover, the data is usually noisy with unavoidable and unpredictable situations such as terrible weather conditions and traffic accidents, which makes the task more challenging.

Various efforts have been devoted to the specific problem of trajectory regression. The method proposed in [3] adopts Gaussian process regression to predict the cost of a complete trajectory. The work in [4] essentially enforces spatial smoothness over links and can be modeled as a simple form of kernel ridge regression. However, there are noticeable limitations in these two approaches where the former method only deals with the situation that all trajectories share the same origin and destination while the latter one ignores the fact that the cost of a road segment fluctuates smoothly most of the time with a few transitions. Taking this dynamic factor into consideration, a multi-task learning approach is proposed in [5] to incorporate temporal dynamics, which treats each time period as a task and introduces a cross-task regularization term encouraging smoothly changing costs of successive time slots.

However, there are still issues that remain unaddressed. Although the link costs fluctuate smoothly most of the time during a day, there exist noticeably abrupt rises in the values of traffic costs during rush hours with daily occurrence, ignoring which may result in biased models with sensitivity to traffic peaks. In the proposed framework, we formulate the problem via a multi-task framework, which is known as a technique to improve the generalization performance by leveraging the intrinsic relationships among tasks [6]–[12].

Specifically, we represent the multiple regression weights for the temporal tasks via a matrix, where each column corresponds to a task (time slot), and each row to a feature (road segment). We first decompose the weight matrix into a sum of two components, then penalize the variance of regression weights of the tasks in the first component explicitly, following the assumption that the costs of road segments generally change smoothly during off-peak periods. Next a graph regularization term is adopted on the second component as in [4] to enforce the spatial proximity and the smoothness over road segments. Additionally, the block sparse $\ell_{\infty,1}$ norm is employed on columns/tasks of the third component, which can help to identify the tasks with significant temporal changes in rush hours and reduce the influence of outlier tasks on the learned model. An

1550-4786/14 $31.00 © 2014 Crown Copyright
DOI 10.1109/ICDM.2014.132

accelerated proximal algorithm is designed to solve the convex but non-smooth optimization problem efficiently. The proposed method is evaluated experimentally on both synthetic and real data sets with convincing results.

The rest of the paper is organized as follows. In Section III we present the formulated MTL-based framework. An accelerated proximal algorithm is then proposed in Section IV to solve the optimization problem, followed by the experimental results on both synthetic and real data sets in Section V. The paper is then concluded in Section VI.

II. Problem Definition

This section summarizes the trajectory regression problem with relevant background. We start with the definitions in road networks following [3].

Definition 1 (LINK). *A link is a road segment between two neighbor intersections.*

Definition 2 (TRAJECTORY). *A trajectory is a sequence of links, where any two consecutive links share an intersection.*

We are given a set of N trajectory-cost pairs on a road network consisting of d links,

$$\mathcal{D} \equiv \{(\mathbf{x}_i, y_i) | i = 1, 2, \ldots, N\}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ represents the i-th trajectory with each feature corresponding to a link, and y_i is the observed travel time over the trajectory $\mathbf{x}_i$. Specifically, the k-th entry in $\mathbf{x}_i$ is the distance actually traveled along the k-th link if the trajectory has passed that link, and 0 otherwise. Now the goal is to learn the weights $\mathbf{w} \in \mathbb{R}^d$ and then predict the total travel cost for an arbitrary trajectory.

III. Robust Dynamic Trajectory Regression with Multi-Task Learning

Intuitively, the traffic costs of road segments are not static over different time periods. To model the temporally changing link costs, we can divide $\mathcal{D}$ into m disjoint subsets ordered by time: $\mathcal{D} = \mathcal{D}^1 \cup \mathcal{D}^2 \cdots \cup \mathcal{D}^m$, $\mathcal{D}^{t_1} \cap \mathcal{D}^{t_2} = \emptyset$ $(t_1, t_2 = 1, 2, \ldots, m, t_1 \neq t_2)$, where t_i represents the t_i-th time period.

By assuming the Gaussian noise σ_t for the t-th task, we have $p(y_i^t | \mathbf{x}_i^t) = N(y_i^t | \mathbf{w}^t \cdot \mathbf{x}_i^t, \sigma_t^2)$, the problem can then be modeled by maximizing the likelihood combining all the tasks and formulated as

$$\min_W \mathcal{L}(W) = \min_W \sum_{t=1}^m \|Y^t - X^t \mathbf{w}^t\|_2^2 \tag{1}$$

where $Y^t = [y_1^t, \ldots y_{n_t}^t]^\top \in \mathbb{R}^{n_t \times 1}$, $X^t = [\mathbf{x}_1^t, \ldots \mathbf{x}_{n_t}^t]^\top \in \mathbb{R}^{n_t \times d}$, with n_t denoting the number of samples in $\mathcal{D}^t$, $\sum_t^m n_t = N$.

Note that the bias term is left out in the model since intuitively the cost of any trajectory that spans no links will be 0. All parameters across the entire time domain are integrated in the matrix $W = [\mathbf{w}^1, \mathbf{w}^2, ..., \mathbf{w}^m] \in \mathbb{R}^{d \times m}$ where the t-th column represents the time cost per distance unit for each link during the t-th time period, while the k-th row describes the dynamic fluctuations of the cost on link k. Thus, W captures both temporal and spatial heterogeneity over the road network with its columns and rows respectively.

A. The Additive Model for Multi-Task Learning

In the specific context of trajectory regression, the amount of training data of trajectory-cost pairs is generally limited. As a consequence, it is necessary to leverage the relationships between the temporal tasks to alleviate the sparsity problem for individual tasks.

In multi-task learning literature, additive models [13]–[15] exploit multiple forms of relationships among tasks, and fit naturally with the trajectory regression problem where we want to capture the temporal and spatial variations simultaneously. Specifically, following the idea of additive models, in the proposed framework the weight matrix W is decomposed into two components $W = P + Q$ where P models the global similarities over links and time, and Q captures the outliers including rush hours.

B. Global Temporal and Spatial Regularization

1) Global Temporal Smoothness: From the global perspective, the link costs change smoothly most of the time. Therefore we penalize the variance of regression weights of the tasks with the following regularization term:

$$\mathrm{tr}(PL_1P^\top) = \sum_{t=1}^m \|P_{:,t} - \frac{1}{m}\sum_{r=1}^m P_{:,r}\|_2^2 \tag{2}$$

where $P_{:,t}$ represents the t-th column of P, and $L_1 = I - \frac{1}{m}\mathbf{1}\mathbf{1}^\top$. The regularization term (2) calculates the sum of the element-wise variance of $P_{:,1}, P_{:,2}, \ldots, P_{:,m}$, enforcing the columns of P or the tasks to be similar with some discrepancy.

2) Global Spatial Smoothness: We adopt a graph regularization term to enforce spatial smoothness following the recipe in [4]:

$$\mathrm{tr}(P^\top L_2 P) = \sum_{i,j=1}^d S_{ij} \|P_{i,:} - P_{j,:}\|_2^2 \tag{3}$$

where $P_{i,:}$ represents the i-th row of P and S is the similarity matrix of links and can be calculated as in [4]. Essentially, L_2 is the Laplacian matrix of the link graph, with the effect of enforcing spatial smoothness, and can be calculated as $(L_2)_{i,j} = \delta_{ij}\sum_{k=1}^d S_{ik} - S_{ij}$ where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise.

C. Temporal Block Sparse Structure

In addition to the global smoothness, Q is designed to model the non-global patterns and capture the outliers. One can observe that the significant temporal transitions in traffic usually appear in a few specific time periods densely with large values in link costs. On the other hand, the other type of outliers, which are caused by data contamination and measurement errors, usually appear randomly, which should be zeroed out. The two factors essentially imply simultaneously dense columns and a block sparse structure in Q as shown in Fig. 1.

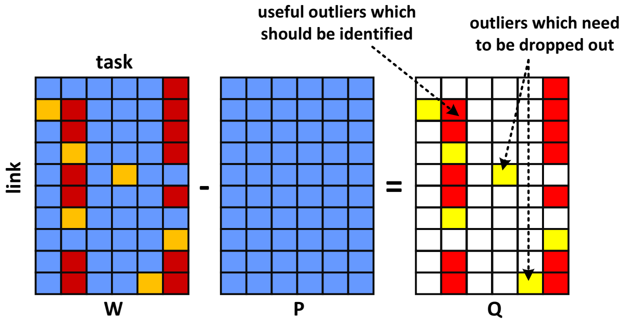
useful outliers which should be identified
outliers which need to be dropped out
task
link
-
=
W
P
Q

Figure 1. Two types of outliers in Q: the useful patterns of peak traffic (red) to be identified, and the random outliers (yellow) to be removed.

To enforce a temporal block sparse structure on Q, we adopt the $\ell_{p,1}$ norm as regularization which is convex [16] and known as "group lasso". Specifically, for any matrix Z, the $\ell_{p,1}$ norm can be defined as $\|Z\|_{p,1} = \sum_j \|Z_{:,j}\|_p$ where $Z_{:,j}$ is the j-th column of Z. The $\ell_{p,1}$ regularization with $p > 1$ is known to facilitate block sparsity.

Here we set $p = \infty$ which results in the $\ell_{\infty,1}$ norm defined as $\|Z\|_{\infty,1} = \sum_j \|Z_{:,j}\|_\infty$, where $\|Z_{:,j}\|_\infty = \max_i |Z_{ij}|$. In our case, this regularization term helps to enforce column sparsity to identify the useful patterns of peak traffic while removing the second type of outliers. Moreover, as mentioned in [17], the $\ell_{\infty,1}$ norm intuitively contributes to favorable recovery of the non-zero elements in non-zero columns. Compared to the $\ell_{2,1}$ regularization in which the ℓ_2 norm of all non-zero columns are summed up and all non-zero elements contribute in those columns, the $\ell_{\infty,1}$ norm is only influenced by the maximum elements of the non-zero columns. This agrees with the nature of the trajectory regression problem better: the cost of a trajectory is mostly decided by the link with highest cost during traffic peaks.

Based on the above discussion, we integrate global spatial and temporal smoothness as well as significant dynamic changes, and the robust dynamic multi-task trajectory regression (RDMTR) framework can be formulated as:

$$\begin{aligned}\min_{W=P+Q,\, Q\geq 0} \quad & \sum_{t=1}^{m} \left\|Y^t - X^t \mathbf{w}^t\right\|_2^2 + \lambda_1 \mathrm{tr}(P L_1 P^\top) \\ & + \lambda_2 \mathrm{tr}(P^\top L_2 P) + \lambda_3 \|Q\|_{\infty,1} \qquad (4)\end{aligned}$$

D. Related Work

Our work is an extension of the static RETRACE model in [4]. Interestingly, another extension of [4] is proposed in [5] as the DTRTS method in which a variant of fused lasso regularization term $\Omega_{\mathrm{DTRTS}}(W) = \sum_{i=1}^{d} \left(\sum_{t=2}^{m} |W_{i,t} - W_{i,t-1}|\right)^2$ is employed to model the dynamic changes. However it ignores the outliers caused by the rush hours and corrupted data as shown in Fig.1, which implies sensitivity of the DTRTS model.

IV. Optimization

Problem (4) is convex, but the $\ell_{\infty,1}$ regularization term is not trivial for optimization due to its non-smoothness and mixed-norm structure. In this section, we leverage the well-developed proximal algorithm to solve it effectively.

A. Proximal Methods

Proximal gradient algorithms have received significant theoretical and empirical success in various problems [18], [19]. Formally, proximal methods are capable of solving the general form of optimization as:

$$\min_{Z} \{F(Z) + R(Z)\} \qquad (5)$$

where both $F(Z)$ and $R(Z)$ are both convex; $F(Z)$ is differentiable but $R(Z)$ is non-smooth. Denoting $Z = \begin{pmatrix} P \\ Q \end{pmatrix}$, we observe that (4) is exactly a specific form of (5) where $F(Z) = \ell(W) + \lambda_1 \mathrm{tr}(P L_1 P^\top) + \lambda_2 \mathrm{tr}(P^\top L_2 P)$ and $R(Z) = \lambda_3 \|Q\|_{\infty,1}$.

Proximal algorithms adopt the iterative scheme. Specifically, given the previous estimate Z_{r-1}, Z_r can be updated with linear approximation of $F(Z_r)$ at Z_{r-1} in the r-th iteration. Next, by adding $R(Z)$ and we get

$$Z_r = \arg\min_{Z} \frac{\gamma_r}{2} \|Z - C_Z(Z_{r-1})\|_F^2 + R(Z) \qquad (6)$$

in which $C_Z(Z_{r-1})$ is a constant with regard to the previous point Z_{r-1}. Notice γ_r is a positive real number representing the step size. By decoupling P and Q we get two subproblems:

$$P_r = \arg\min_{P} \frac{\gamma_r}{2} \|P - C_P(P_{r-1})\|_F^2 \qquad (7)$$

$$Q_r = \arg\min_{Q} \frac{\gamma_r}{2} \|Q - C_Q(Q_{r-1})\|_F^2 + \lambda_3 \|Q\|_{\infty,1} \qquad (8)$$

where $C_P(P_{r-1}) = P_{r-1} - \bigtriangledown F_P(P_{r-1})/\gamma_r$, $C_Q(Q_{r-1}) = Q_{r-1} - \bigtriangledown F_Q(Q_{r-1})/\gamma_r$ and $\bigtriangledown F_Z(C)$ represents the gradient of $F(Z)$ with regard to Z at point C.

Before describing the procedures of solving the two subproblems (7) and (8), it is necessary to note that the proximal framework provides a scheme to estimate the step size properly, which essentially works by increasing γ_r while the inequality below is satisfied.

$$\begin{aligned}F(Z_r) \leq F(Z_{r-1}) + \frac{\gamma_r}{2} \|Z_r - Z_{r-1}\|_F^2 \\ + \langle Z_r - Z_{r-1}, \bigtriangledown F_Z(Z_{r-1}) \rangle \end{aligned} \qquad (9)$$

B. Computing P

A closed-form solution of problem (7) can be easily obtained as $P_r = C_P(P_{r-1})$.

C. Computing Q

Now we consider the problem (8). By leveraging the Moreau Decomposition, we get the following proposition:

Proposition 1. *Given $C \in R^{a\times b}$, the objective function*

$$\min_{Q} \frac{1}{2}\|Q - C\|_F^2 + \lambda\|Q\|_{\infty,1} \tag{10}$$

has a closed-form solution, which can be obtained by

$$Q_{i,j} = \begin{cases} \lambda, & C_{i,j} > \lambda \\ C_{i,j}, & -\lambda \le C_{i,j} \le \lambda \\ -\lambda, & C_{i,j} < -\lambda \end{cases} \tag{11}$$

Proof: Denote $\mathbf{q}^i$ and $\mathbf{c}^i$ the i-th column of Q and C respectively, Problem (10) can be decomposed into b subproblems in terms of each column:

$$\min_{\mathbf{q}^i} \frac{1}{2}\|\mathbf{q}^i - \mathbf{c}^i\|_2^2 + \lambda\|\mathbf{q}^i\|_\infty \tag{12}$$

With the Moreau Decomposition Property [19], we have $\mathbf{c} = \text{prox}_R(\mathbf{c}) + \text{prox}_{R^*}(\mathbf{c})$, where R^* is the conjugate form of R. Note that the conjugate form of the ℓ_∞ norm is the ℓ_1 norm. Therefore, $\mathbf{q}^i$ could be obtained by solving the following

$$\mathbf{q}^i = \min_{\mathbf{q}^i} \left\{ \mathbf{c}^i - \left(\frac{1}{2}\|\mathbf{q}^i - \mathbf{c}^i\|_2^2 + \lambda\|\mathbf{q}^i\|_1 \right) \right\} \tag{13}$$

and sequentially we get (11).█

Following the accelerated scheme by Nesterov [18], the algorithm for solving Problem (4) is detailed in Algorithm 1.

D. Time Complexity

Denoting the average number of training samples of tasks by $\bar{n}$, the per-iteration time complexity of the proposed RDMTR is $O(\bar{n}dm)$ since the closed-form solutions only involve linear operations. Meanwhile, by solving a subproblem of quadratic programming for each decoupled row of W in every iteration, the per-iteration time complexity of DTRTS [5] is raised to $O(\bar{n}d^2m^3)$, which is much higher.

V. Experiments

Experiments are conducted to evaluate the proposed RDMTR on both synthetic and real-world data sets.

A. Competing Algorithms and Measurements

To verify the effectiveness of the proposed RDMTR, Ridge (ridge regression without splitting the task) and STL-Ridge (ridge regression conducted on individual tasks separated by time slots) are treated as baselines. Similarly, RETRACE and STL-RETRACE represent RETRACE [4] with one task and multiple single tasks respectively. Other comparable algorithms include DTRTS in [5], TGL in [11], Dirty in [15] and RDMTR$_{2,1}$ which adopts $\ell_{2,1}$ norm alternative of the block sparse norm in (4) .

Algorithm 1 Accelerated Proximal Method for RDMTR

```
Input: $X^1, ..., X^m$; $Y^1, ..., Y^m$; $\lambda_1$; $\lambda_2$; $\lambda_3$.
Output: $W$.
Initialize: $P_0$, $Q_0$, $P_{-1}$, $Q_{-1}$, $\gamma_0$,
    $L > 1$, $t_0 = 0$, $t_1 = 1$ , $r = 1$;
repeat
    $\alpha = \frac{t_{r-1}-1}{t_r}$;
    $P = (1+\alpha)P_{r-1} - \alpha P_{r-2}$ ;
    $Q = (1+\alpha)Q_{r-1} - \alpha Q_{r-2}$;
    $\gamma_r = \gamma_{r-1}$;
    while true do
        $P_r = C_P(P)$;
        Update $Q_r$ by (11);
        Set the negative elements of $Q_r$ to be 0s;
        $\gamma_r = \gamma_r L$;
        if (9) is satisfied:
            break;
    end while
    $t_{r+1} = \frac{1+\sqrt{1+4t_r^2}}{2}$;
    $r = r + 1$;
until convergence
$W = P_r + Q_r$;
```

The parameters for all the competing algorithms are adjusted with 3-fold cross validation. It should be mentioned that the parameter which controls the identical spatial smoothness regularization in RETRACE, STL-RETRACE, DTRTS and the proposed RDMTD are set to the same value for fairness. The maximum number of iterations is set to 1500 while the tolerance is 10^{-5}. All results are averaged over 10 repetitions.

Algorithms are evaluated with measurements including the normalized mean squared error (nMSE) and the averaged mean squared error (aMSE) as defined in [14]. Notice that smaller values of nMSE and aMSE represent better regression performance.

B. Synthetic Data - Grid20

The synthetic experimental setup follows similar design as in previous work [4]. A 20×20 two-dimensional lattice structure is generated with 760 500-meter-long links. An individual task corresponds to each hour in a day and we get 24 tasks in total. For each task $\mathbf{p}^i$ is generated with uniform distribution $\mathcal{U}(\mathbf{0}, \mathbf{0}.5)$ and then smoothed according to the topological structure with a convolution operator. To keep tasks similar with discrepancy, we set $\mathbf{p}^i = \alpha_i \mathbf{p}^i$ with α_i chosen randomly between 1.0 and 1.1, and 3 time periods are randomly selected as peak hours with extra costs $\mathbf{q}^i \sim \mathcal{N}(\mathbf{1}, \mathbf{10})$. Assuming that not all links suffer from heavy traffic during the peak hours, 70% of the elements of $\mathbf{q}^i$ in the peak hours are selected randomly and set to 0; finally the i-th weight vector is calculated as $\mathbf{w}^i = \mathbf{p}^i + \mathbf{q}^i$.

Further, 400 samples are generated for each task, by finding the shortest routes with the origin points and destination points selected randomly, and the trajectory-cost pairs are generated by $Y^i = (X^i\mathbf{w}^i) * \mathcal{N}(\mathbf{1}, \mathbf{5})$, where $\mathcal{N}(\mathbf{1}, \mathbf{5})$ is the noise factor. Then $10\%, 20\%$, and 30% of the samples from each task are randomly selected as the training set respectively, the other 20% as the validation set, and the remaining are used as the test set. Averaged nMSE and aMSE values after 10 repetitions are reported in Table I.

The purpose of this experiment is to evaluate the abilities of capturing temporal changes of the 9 algorithms. Overall, the proposed RDMTR and alternative RDMTR$_{2,1}$ perform the best among all the algorithms, where RDMTR surpasses RDMTR$_{2,1}$ with training ratios 20% and 30%. Interestingly, in this scenario, STL-RETRACE outperforms RETRACE, implying that the discrepancies among tasks overrule the similarities. As a result, TGL, Dirty and DTRTS more or less suffer from negative information transfer among tasks.

C. Suzhou Traffic Data

The Suzhou Traffic Data contains 59593 trajectory records of 4797 taxies from 7:00 to 19:59 in urban area of Suzhou, China during the first week in March, 2012. Trajectories occurring in the downtown region which covers 2105 road segments and about 100 km^2 centered at $(E120°37', N31°19')$ are recorded. As illustrated in Fig. 2, the traffic load for each link is extremely unbalanced. After splitting the data by each hour, 13 regression tasks are entailed.

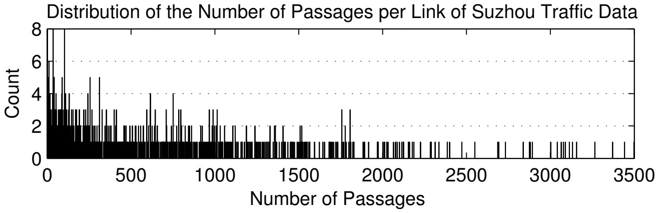

Figure 2. The distribution of the number of passages per link in Suzhou.

We randomly select 20%, 30%, and 40% of the samples from each task as the training set and 20% as the validation set, leaving the rest as the test set. The results after 10 repetitions are reported in Table II, which demonstrates that the proposed RDMTR approach outperforms the other algorithms including the alternative RDMTR$_{2,1}$ in terms of nMSE and aMSE. Interestingly, compared to Ridge and RETRACE with no separate tasks, most multi-task algorithms including Dirty and DTRTS are not superior in performance, which implies negative information transfer among tasks.

To show the comparison more clearly, we further plot the curves of 6 representative algorithms in nMSE and aMSE with different training ratios, which again demonstrate the clear advantage of the RDMTR approach.

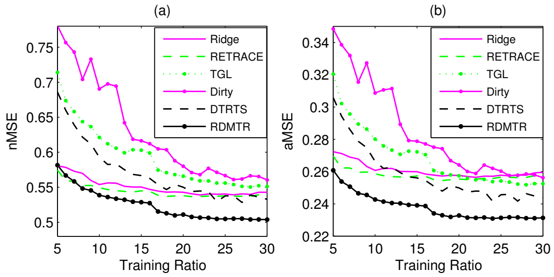

Figure 3. Comparison of different algorithms in terms of nMSE and aMSE.

Moreover, to verify the ability of Q capturing the outlier costs, we calculate the average of Q over 10 repetitions and plot the maximum entry of each column in Fig. 4. From the curves we can observe two peaks at time slots 8 and 17, which correspond to significantly higher traffic costs in rush hours around 8:00–8:59 and 17:00–17:59.

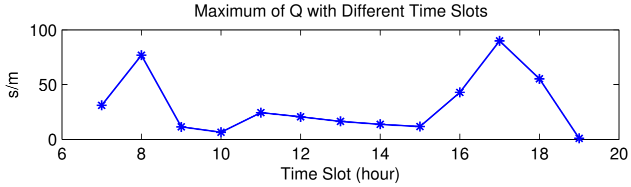

Figure 4. Maximum value of Q w.r.t. different time slots.

VI. Conclusion

This paper focuses on the "trajectory regression" problem. We propose a multi-task framework that divides the problem into two components to model the spatial and temporal structure globally as well as the local patterns of traffic costs in rush hours. Experiments on both synthetic and real data sets demonstrate the clear advantage of the proposed framework over the other state-of-the-art approaches. In future work we would like to consider the issue of incomplete data in the trajectory regression problem and explore further into the inherent structures of the problem.

Acknowledgment

Research supported by the National Natural Science Foundation of China (No. 61375060) and the Fundamental Research Funds for the Central Universities (WK0110000036).

References

[1] G. S and S. P, "Trajectory clustering with mixtures of regression models," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining,ACM*, 1999, pp. 63–72.

[2] V. M, K. G, and G. D, "Elastic translation invariant matching of trajectories," *Machine Learning*, vol. 58, no. 2-3, pp. 301–334, 2005.

[3] I. T and K. S, "Travel-time prediction using gaussian process regression: A trajectory-based approach," in *SDM*, 2009, pp. 1183–1194.

Table I
PERFORMANCE OF THE COMPETING ALGORITHMS IN TERMS OF nMSE AND aMSE ON THE SYNTHETIC DATA. METHODS WITH THE BEST AND COMPARABLE PERFORMANCE (PAIRED T-TESTS AT 95% SIGNIFICANCE LEVEL) ARE BOLDED.

Training Ratio	10%(nMSE)	20%(nMSE)	30%(nMSE)	10%(aMSE)	20%(aMSE)	30%(aMSE)
Ridge	0.978 ± 0.008	0.959 ± 0.011	0.948 ± 0.009	0.760 ± 0.006	0.745 ± 0.008	0.737 ± 0.006
STL-Ridge	1.133 ± 0.032	0.621 ± 0.026	0.427 ± 0.013	0.415 ± 0.012	0.231 ± 0.008	0.159 ± 0.004
RETRACE	0.986 ± 0.011	0.971 ± 0.018	0.958 ± 0.009	0.766 ± 0.008	0.755 ± 0.013	0.745 ± 0.006
STL-RETRACE	0.599 ± 0.006	0.548 ± 0.005	0.510 ± 0.008	0.230 ± 0.002	0.211 ± 0.003	0.196 ± 0.004
TGL	1.009 ± 0.030	0.562 ± 0.019	0.396 ± 0.011	0.367 ± 0.011	0.207 ± 0.006	0.146 ± 0.002
Dirty	1.087 ± 0.033	0.595 ± 0.025	0.418 ± 0.013	0.399 ± 0.012	0.221 ± 0.008	0.156 ± 0.003
DTRTS	1.215 ± 0.078	0.717 ± 0.040	0.547 ± 0.043	0.447 ± 0.030	0.268 ± 0.015	0.206 ± 0.004
RDMTR$_{2,1}$	**0.567 ± 0.012**	0.454 ± 0.007	0.391 ± 0.017	**0.217 ± 0.005**	0.174 ± 0.004	0.150 ± 0.004
RDMTR	0.603 ± 0.013	**0.443 ± 0.004**	**0.358 ± 0.007**	0.222 ± 0.004	**0.166 ± 0.004**	**0.134 ± 0.003**

Table II
PERFORMANCE OF VARIOUS ALGORITHMS IN TERMS OF nMSE AND aMSE ON SUZHOU TRAFFIC DATA. METHODS WITH THE BEST AND COMPARABLE PERFORMANCE (PAIRED T-TESTS AT 95% SIGNIFICANCE LEVEL) ARE BOLDED.

Training Ratio	20%(nMSE)	30%(nMSE)	40%(nMSE)	20%(aMSE)	30%(aMSE)	40%(aMSE)
Ridge	0.549 ± 0.013	0.547 ± 0.019	0.540 ± 0.025	0.259 ± 0.006	0.257 ± 0.012	0.254 ± 0.019
STL-Ridge	0.617 ± 0.027	0.589 ± 0.040	0.560 ± 0.029	0.279 ± 0.013	0.265 ± 0.014	0.253 ± 0.021
RETRACE	0.546 ± 0.013	0.545 ± 0.022	0.538 ± 0.027	0.258 ± 0.007	0.256 ± 0.014	0.253 ± 0.020
STL-RETRACE	0.668 ± 0.035	0.628 ± 0.042	0.587 ± 0.036	0.300 ± 0.016	0.281 ± 0.017	0.266 ± 0.025
TGL	0.570 ± 0.032	0.581 ± 0.063	0.538 ± 0.046	0.256 ± 0.013	0.256 ± 0.022	0.240 ± 0.021
Dirty	0.626 ± 0.034	0.615 ± 0.056	0.590 ± 0.034	0.282 ± 0.017	0.275 ± 0.019	0.265 ± 0.021
DTRTS	0.612 ± 0.051	0.596 ± 0.048	0.531 ± 0.022	0.277 ± 0.026	0.266 ± 0.022	0.239 ± 0.015
RDMTR$_{2,1}$	0.525 ± 0.024	0.562 ± 0.068	0.525 ± 0.049	0.241 ± 0.010	0.249 ± 0.023	0.239 ± 0.021
RDMTR	**0.494 ± 0.014**	**0.498 ± 0.040**	**0.481 ± 0.023**	**0.224 ± 0.006**	**0.223 ± 0.015**	**0.217 ± 0.015**

[4] I. T and S. M, “Trajectory regression on road networks,” in *AAAI*, 2011.

[5] Z. J and N. L. M, “Time-dependent trajectory regression on road networks via multi-task learning,” in *AAAI*, 2013.

[6] A. Argyriou, T. Evgeniou, and M. Pontil, “Convex multi-task feature learning,” *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[7] T. Evgeniou and M. Pontil, “Regularized multi-task learning,” in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’04, 2004, pp. 109–117.

[8] L. Jacob, F. Bach, and J.-P. Vert, “Clustered multi-task learning: A convex formulation.” in *Advances in Neural Information Processing Systems*, 2008, pp. 745–752.

[9] Y. Zhang and D.-Y. Yeung, “A convex formulation for learning task relationships in multi-task learning,” in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010, pp. 733–442.

[10] K. S and X. E. P, “Tree-guided group lasso for multi-task regression with structured sparsity,” in *Proceedings of the 27th International Conference on Machine Learning(ICML)*, 2010, pp. 543–550.

[11] Z. J, Y. L, L. J, and Y. J, “A multi-task learning formulation for predicting disease progression,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 814C–822.

[12] Z. J, L. J, Narayan, and A. V, “Modeling disease progression via fused sparse group lasso,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012.

[13] P. Gong, J. Ye, and C. Zhang, “Robust multi-task feature learning,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’12, 2012, pp. 895–903.

[14] J. Chen, J. Zhou, and J. Ye, “Integrating low-rank and group-sparse structures for robust multi-task learning,” in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 42–50.

[15] J. A, R. P. D, and S. S, “A dirty model for multi-task learning.” in *Advances in Neural Information Processing Systems*, 2010.

[16] Y. M and L. Y, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[17] S. Negahban and M. J. Wainwright, “Joint support recovery under high-dimensional scaling: Benefits and perils of $l_{1,\infty}$-regularization,” in *In Advances in Neural Information Processing Systems (NIPS)*, 2008.

[18] Y. Nesterov, “Gradient methods for minimizing composite objective function,” *Technical Report,Catholic University*, 2007.

[19] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 3, no. 1, pp. 123–231, 2014.