Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Time-Dependent Trajectory Regression on Road Networks via Multi-Task Learning

Jiangchuan Zheng[♮] and **Lionel M. Ni**[♮,♯]
[♮]Department of Computer Science and Engineering
[♯]Guangzhou HKUST Fok Ying Tung Research Institute
Hong Kong University of Science and Technology
Hong Kong, China
{jczheng, ni}@cse.ust.hk

Abstract

Road travel costs are important knowledge hidden in large-scale GPS trajectory data sets, the discovery of which can benefit many applications such as intelligent route planning and automatic driving navigation. While there are previous studies which tackled this task by modeling it as a regression problem with spatial smoothness taken into account, they unreasonably assumed that the latent cost of each road remains unchanged over time. Other works on route planning and recommendation that have considered temporal factors simply assumed that the temporal dynamics be known in advance as a parametric function over time, which is not faithful to reality. To overcome these limitations, in this paper, we propose an extension to a previous static trajectory regression framework by learning the temporal dynamics of road travel costs in an innovative non-parametric manner which can effectively overcome the temporal sparsity problem. In particular, we unify multiple different trajectory regression problems in a multi-task framework by introducing a novel cross-task regularization which encourages temporal smoothness on the change of road travel costs. We then propose an efficient block coordinate descent method to solve the resulting problem by exploiting its separable structures and prove its convergence to global optimum. Experiments conducted on both synthetic and real data sets demonstrate the effectiveness of our method and its improved accuracy on travel time prediction.

Introduction

Route planning services, which recommend fastest and most efficient routes to drivers, are growing in wide popularity in recent years.These services have clear advantages as they help facilitate energy-efficient driving, save drivers' time, and relieve traffic load on city transportation systems.

Although fastest route search can be viewed as a shortest path finding problem, existing shortest path finding algorithms cannot be directly used for two reasons. First, the lengths of road segments cannot be straightforwardly used as edge weights, as travel costs are decided by many factors such as road classes and traffic conditions. Second, the edge weights are by no means static as driving speed patterns may present periodically recurring temporal dynamics, which varies from road to road, as evidenced by (Yuan et al. 2010). For example, downtown areas usually have lower speed (higher travel cost) during rush hours than usual, while speed patterns in suburbs remain relatively constant.

Existing works addressing the fastest route search problem can be divided into two categories. On one hand, there are works that infer latent travel costs of road segments by mining historical trajectory data, e.g, (Idé and Sugiyama 2011) formulated a spatial proximity-regularized trajectory regression problem to learn the potential cost of each road from trajectory data; (Idé and Kato 2009) applied Gaussian Process to directly predict the cost of a whole trajectory by exploiting its similarity with trajectories in the training set. However, these works unreasonably assumed static travel costs on each road, while ignoring the potential temporal dynamics. On the other hand, there are works that integrated the temporal dynamics of road travel costs into the computation of the fastest route such as (Kanoulas et al. 2006). However, they assumed that the temporal dynamics on each road be known in advance as a piecewise constant function, which is unrealistic in practice. In a word, the works in these two categories complement each other, but no previous work has well leveraged both of their advantages in a principled and unified framework.

In view of this gap, we propose a novel optimization framework in this paper which can not only learn the latent travel costs of road segments from historical trajectory data but also accounts for temporal dynamics of those costs in an innovative non-parametric manner, i.e, without assuming the function form of the temporal dynamics in advance. This learning task can facilitate many applications such as route travel time prediction, city traffic dynamics monitoring, etc. The only assumption we make is that the temporal dynamics of a road shows relatively stable periodic recurring patterns over days, which is acceptable in real world traffic.

Without assuming the function form of the temporal dynamics in advance, a straightforward approach to our problem would be to separate the time domain into a series of fine-grained time slots and then learn costs of roads in each time slot using only the trajectories falling in that slot. However, this simple method would suffer from severe overfitting caused by the issue of temporal sparsity. In particular, although large in scale, a trajectory data set usually presents a skewed temporal distribution, i.e, a number of time slots might have few trajectories recorded. Moreover, the travel

costs on different roads might change in different ways over time, making the problem even more challenging.

To tackle these challenges, we propose, based on the static trajectory regression framework proposed by (Idé and Sugiyama 2011), a multi-task learning framework with a novel temporal regularization term that allows information sharing between neighboring time slots. On one hand, by treating each time slot as a separate learning task, the framework accounts for road cost change over time, which essentially learns temporal dynamics in a non-parametric manner. On the other hand, with an appropriate information sharing between neighboring time slots, the framework effectively overcomes the temporal sparsity challenge, thus revealing meaningful temporal dynamic patterns. The intuition is that, the latent cost of a road usually changes smoothly with apparent "flat intervals" rather than fluctuating arbitrarily. For example, the "rush hour" represents a flat interval and the "working hour" represents another. The latent cost in each interval remains relatively constant but different intervals might show apparently different costs. To model this intuition, we introduce a cross-task regularization that employs the idea of fused lasso (Tibshirani et al. 2005) to encourage sparsity in temporally successive cost differences as a natural formulation of the observation that there are usually only a few cost "transitions" along the time domain. In this way, we essentially address the temporal sparsity challenge by allowing a smooth information flow along the time domain. From an overall perspective, our framework, which favors temporal smoothness, learns for each road how to "partition" the time domain into a few intervals with each interval taking one relatively unique constant cost. As we shall see later, although the resulting learning problem is proved convex, it is non-differentiable and has too many parameters, making it challenging to solve. To address this challenge, we propose a block-coordinate descent method which can not only solve the problem efficiently by exploiting its separable structures, but also is guaranteed to converge to the global optimum. We summarize our contributions as follows:

1) We propose the problem of learning temporal dynamics of travel costs on road segments from trajectory data sets, which can benefit many smart-city applications such as fastest route recommendations and city traffic monitoring.

2) We propose a multi-task learning framework in time domain with a novel regularization encouraging temporal smoothness which can not only uncover the temporal dynamics of road costs in a non-parametric manner, but also overcome the temporal sparsity in trajectory data.

3) We propose an efficient block-coordinate descent algorithm to solve the learning problem which is guaranteed to converge to the global optimum.

4) We perform extensive experiments on both synthetic and real data sets to demonstrate the effectiveness and predictive power of our method.

Problem Formulation

In this section, we review the static trajectory regression framework in (Idé and Sugiyama 2011) briefly, and then extend it to a multi-task learning framework in time domain. We use "road segment" and "link" interchangeably.

Static Trajectory Regression Framework

(Idé and Sugiyama 2011) formulated the task of learning static road costs from trajectory data as a regression problem with a penalty encouraging spatial smoothness of costs over neighboring roads. Let $\mathbf{w} \in R^M$ encode the cost per distance unit for each link, where M is the number of links. Suppose there are N trajectories in the data set $\mathcal{D}$, and the nth trajectory is represented as $\boldsymbol{\phi}_n \in R^M$, where its mth element is the length of the mth link if the trajectory has passed that link, and 0 otherwise. The whole trajectory set can therefore be encoded in matrix $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1^T, \boldsymbol{\phi}_2^T, ..., \boldsymbol{\phi}_N^T)^T \in R^{N\times M}$, where the nth row is $\boldsymbol{\phi}_n^T$. Let vector $\mathbf{y} \in R^N$ record the total time spent on each trajectory, then $\mathbf{w}^T\boldsymbol{\phi}_n$ is a good approximation of $\mathbf{y}_n$ because the total time spent is the sum of the contributions on the individual links. By assuming a Gaussian noise model

$$p(\mathbf{y}_n|\boldsymbol{\phi}_n) = \mathcal{N}(\mathbf{y}_n|\mathbf{w}^T\boldsymbol{\phi}_n, \sigma^2) \tag{1}$$

the total loss to be minimized for learning $\mathbf{w}$ is $||\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}||_2^2$. To overcome spatial sparsity, i.e, to infer costs of roads with few passage history, (Idé and Sugiyama 2011) adds a penalty

$$\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} S_{ij}|\mathbf{w}_i - \mathbf{w}_j|^2 \tag{2}$$

to impose a soft constraint that spatially close roads tend to have similar costs, where S_{ij} is the spatial proximity between link i and link j. (2) can be rewritten as $\mathbf{w}^T\mathcal{L}\mathbf{w}$, where $\mathcal{L}$ is the Laplacian matrix defined as $\mathcal{L}_{ij} = \delta_{ij}\sum_{k=1}^{M} S_{ik} - S_{ij}$. The final optimization problem is:

$$\min_{\mathbf{w}} \ ||\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}||_2^2 + \lambda\mathbf{w}^T\mathcal{L}\mathbf{w} \tag{3}$$

where λ is the penalty factor. This is clearly a kernel ridge regression problem and can be solved analytically.

Trajectory Regression Framework with Temporal Dynamics

Being unaware of time-dependence, the static framework is not suitable in the presence of temporal dynamics in road costs. To tackle this problem, we propose a multi-task approach to integrate temporal dynamics of road costs into the framework. Instead of using a single vector $\mathbf{w}$ to represent the static link costs, we allow for temporally dynamic link costs by dividing the entire time domain (usually a day) into K fine-grained slots ordered by time, and introduce a separate cost vector for each time slot. Let vector $\mathbf{w}_k \in R^M$ encode the cost per distance (unit cost) of each link in time slot k. All parameters can therefore be represented as a $M \times K$ matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K)$. The mth row $\mathbf{w}^m$ describes how the cost on link m changes in a day. $\mathbf{W}$ captures both temporal variation and spatial variation by its columns and rows, respectively. This modeling is motivated by two concerns. First, as there is no domain knowledge for the function form of the cost temporal dynamics, the only way that is faithful to reality is to represent such temporal dynamics as a discrete cost sequence over time and learn it in a non-parametric manner. Second, as costs on different links might

change in different ways in a day, each link should have its own representation of temporal dynamics.

To derive the objective function, we divide the whole trajectory set $\mathcal{D}$ into $\mathcal{D}_1, ..., \mathcal{D}_K$, where $\mathcal{D}_k$ contains only trajectories whose time spans fall in slot k. Trajectories that span multiple time slots are split accordingly. Let $\mathbf{\Phi}_k \in R^{N_k \times M}$ represent trajectories in $\mathcal{D}_k$ which has N_k trajectories, and let $\mathbf{y}_k \in R^{N_k}$ encode the total travel time on each trajectory in $\mathcal{D}_k$. Due to the low sampling rate problem and the GPS recording noises that often occur in data collection procedure, the travel time over individual links cannot be reliably obtained in most cases, thus we only use the total travel time over each trajectory in the training set so as to neutralize the various noises. The total loss for time slot k is then $||\mathbf{y}_k - \mathbf{\Phi}_k \mathbf{w}_k||_2^2$. Similar to (Idé and Sugiyama 2011), we add a penalty $\mathbf{w}_k^T \mathcal{L} \mathbf{w}_k$ for each time slot to encourage spatial smoothness on link costs, where $\mathcal{L}$ is the Laplacian matrix encoding link spatial proximity. We also add a penalty $||\mathbf{w}_k||_2^2$ to regularize the magnitude of $\mathbf{w}_k$.

We now propose a novel temporal regularization to overcome temporal sparsity problem that exploits the inherent temporal smoothness in traffic flow. Intuitively, the travel cost on a link usually changes smoothly due to the smooth evolution of traffic flow in real world. Thus, a typical cost temporal dynamics usually has only a few "transitions", i.e, it appears in the form of a few intervals where the cost within each interval remains relatively constant. To capture two characteristics of link cost temporal dynamics, that is, keeping smooth within an interval as well as having only a few transitions, we add to the objective the following temporal regularization term, which is a variant of the fused lasso:

$$\sum_{m=1}^{M} \Big(\sum_{k=2}^{K} |\mathbf{w}_k^m - \mathbf{w}_{k-1}^m| \Big)^2 \tag{4}$$

where the scalar $\mathbf{w}_k^m$ is $\mathbf{W}_{mk}$. $\sum_{k=2}^{K} |\mathbf{w}_k^m - \mathbf{w}_{k-1}^m|$ is the l_1 norm on the successive cost differences w.r.t the mth link, known as the fused lasso of $\mathbf{w}^m$. Due to the usage of l_1 norm, this regularization tends to drive most successive cost differences to zeros and retain only a few non-zeros, thus achieving the effect of learning for each link a cost temporal dynamics roughly as a piecewise constant function. Note that we are learning cost temporal dynamics in a non-parametric manner without assuming piecewise constancy in advance; we only identify near-piecewise smoothness as an intuitive characteristics of link cost temporal dynamics and encode it as a soft constraint in the optimization framework. We do not use l_2 norm to penalize successive cost differences as doing so tends to produce "wiggly" cost dynamics which is not robust to observation noises and will potentially hurt generalization performance. In the outer layer, we aggregate all link-specific fused lasso quantities using l_2 norm instead of using l_1 norm again for the concern that l_2 norm shrinks all quantities by the same magnitude, which meets our demand of treating each link equally, while using l_1 norm tends to drive most links to have constant costs over the entire time domain and hence defeats our purpose of tracking temporal dynamics of link costs.

Our final optimization problem is given as follows:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{k=1}^{K} \Big(||\mathbf{y}_k - \mathbf{\Phi}_k \mathbf{w}_k||_2^2 + \alpha \mathbf{w}_k^T \mathcal{L} \mathbf{w}_k + \beta \mathbf{w}_k^T \mathbf{w}_k \Big) \\ & + \lambda \sum_{m=1}^{M} \Big(\sum_{k=2}^{K} |\mathbf{w}_k^m - \mathbf{w}_{k-1}^m| \Big)^2 \end{aligned} \tag{5}$$

where $\mathbf{w}_k$ and $\mathbf{w}^m$ denote the kth column and the mth row of $\mathbf{W}$, respectively. α, β, λ are penalty factors set by tuning. (5) can be viewed as an extension of (3) to a multi-task setting where multiple time slot-specific trajectory regression tasks are coupled by a regularization term encouraging temporal smoothness. The first part in (5) is separated by time slots where each fits link costs to a time slot-specific trajectory data set, whereas the second part is separated by road segments with each encouraging temporal smoothness on one link. However, the whole objective function is not separable, making it challenging to solve, which we shall discuss in the next section.

Learning Algorithm

We now discuss how to solve (5) effectively. Denote the objective in (5) as $f = f_0 + \lambda \sum_{m=1}^{M} f_m$ where $f_0 = \sum_{k=1}^{K} \big(||\mathbf{y}_k - \mathbf{\Phi}_k \mathbf{w}_k||_2^2 + \alpha \mathbf{w}_k^T \mathcal{L} \mathbf{w}_k + \beta \mathbf{w}_k^T \mathbf{w}_k \big)$, $f_m = \big(\sum_{k=2}^{K} |\mathbf{w}_k^m - \mathbf{w}_{k-1}^m| \big)^2$. Despite its convexity, it is challenging to solve because of its high-dimension and the non-differentiable nature of f_m. To tackle this, we adopt a block coordinate descent method which can not only solve it efficiently but also is guaranteed to converge to global optimum.

Block coordinate descent (BCD) (Bertsekas 1999) method partitions variables into blocks and only updates one block with all other blocks fixed at each iteration. The convergence of BCD to global optimum requires that the objective be differentiable as well as satisfy certain convexity-related properties (Bertsekas 1999). If the objective is non-differentiable, it might occur that at certain non-stationary point, no further decrease of the objective can be made by only updating one block (Auslender 1976). However, if the non-differentiable part is separable and the block partitioning agrees with that separable structure, BCD is guaranteed to converge to global minimum provided the objective satisfies certain regularity properties (Tseng 2001). For our problem, since the non-differentiable part in (5) is separated by rows of matrix $\mathbf{W}$, we view each row $\mathbf{w}^m$ as a block and adopt cyclic BCD to solve this problem. Specifically, in each iteration, we update one row of $\mathbf{W}$, say $\mathbf{w}^m$, while keeping other rows fixed at current values by solving subproblem (6):

$$\begin{aligned} \min_{\mathbf{w}^m} \quad & \sum_{k=1}^{K} \big(||\mathbf{y}_{mk} - \mathbf{x}_{mk} \mathbf{w}_k^m||_2^2 + c_{mk} \mathbf{w}_k^m + q_m \mathbf{w}_k^{m2} \big) \\ & + \lambda \big(\sum_{k=2}^{K} |\mathbf{w}_k^m - \mathbf{w}_{k-1}^m| \big)^2 \end{aligned} \tag{6}$$

where $\mathbf{y}_{mk} = \mathbf{y}_k - \mathbf{\Phi}_k^{(-m)} \mathbf{w}_k^{(-m)}$ ($\mathbf{\Phi}_k^{(-m)}$ is $\mathbf{\Phi}_k$ with the mth column removed, and $\mathbf{w}_k^{(-m)}$ is $\mathbf{w}_k$ with the mth element removed), $\mathbf{x}_{mk}$ is the mth column of $\mathbf{\Phi}_k$, $c_{mk} =$

$2\alpha \sum_{m' \neq m} \mathcal{L}_{mm'} \mathbf{w}_k^{m'}$, and $q_m = \alpha \mathcal{L}_{mm} + \beta$. In (6), all $\mathbf{w}^{m'} (m' \neq m)$ are fixed at the current values. We then use the optimal point $\mathbf{w}^{m\star}$ of (6) to update the current $\mathbf{w}^m$ in $\mathbf{W}$ and proceed to update $\mathbf{w}^{m+1}$ in a similar way. Algorithm 1 summarizes the steps for solving problem (5).

Algorithm 1: BCD for solving (5)

Input : $\{\mathbf{\Phi}_k\}_{k=1...K}$, $\{\mathbf{y}_k\}_{k=1...K}$, $\mathcal{L}$, α, β, λ
Output: $\mathbf{W}$

```
Initialize W to W^0 ;
for r=1 to R do // R: total iteration number
    for m=1 to M do
        compute y_mk, x_mk, c_mk, q_m for all k using the
        current W;
        update w^m by solving subproblem (6), that is,
        replace the mth row of W with the optimal
        point w^{m*} of (6);
    end
end
```

There are two challenges to address in this algorithm. First, is it guaranteed that the $\mathbf{W}$ sequence generated converges to the global optimum $\mathbf{W}^\star$ of (5); second, how to solve subproblem (6) which has a non-differentiable and non-separable part f_m to get $\mathbf{w}^{m\star}$.

We firstly prove that Algorithm 1 is guaranteed to converge to the global optimum of (5). Our major idea is to firstly prove that the objective in (5) satisfies certain properties required in (Tseng 2001) by the following lemmas, and then directly make use of Theorem 4.1 in (Tseng 2001) to conclude our desired convergence property.

Lemma 1: Suppose $\mathbf{W}$ is initialized as $\mathbf{W}^0 \in R_{++}^{M \times K}$, and $f(\mathbf{W}^0) < +\infty$, then the sublevel set $\{\mathbf{W} | f(\mathbf{W}) \leq f(\mathbf{W}^0)\}$ is closed and bounded.

Lemma 2: f is convex on $\mathbf{W}$.

By these lemmas, we are ready to show the following:

Theorem 1: The sequence $\{\mathbf{W}^r\}_{r=1...R}$ generated by the block-coordinate descent method in Algorithm 1 converges to the global optimum $\mathbf{W}^\star$ of (5).

The proofs of these lemmas and theorems can be found in appendix. Now we proceed to the method of solving subproblem (6). As its non-differentiable part is not separable, coordinate descent method cannot be used as it may get stuck at non-stationary point. Our method is to convert (6) to an equivalent QP (quadratic program) as in (7) (the proof of the obvious equivalence is omitted to save space):

$$
\begin{aligned}
\min_{\mathbf{w}^m, t, \boldsymbol{\theta}^+, \boldsymbol{\theta}^-} \quad & \sum_{k=1}^{K} \left(||\mathbf{y}_{mk} - \mathbf{x}_{mk} \mathbf{w}_k^m||_2^2 + c_{mk} \mathbf{w}_k^m \right. \\
& \left. + q_m \mathbf{w}_k^{m2} \right) + \lambda t^2 \\
s.t \quad & \mathbf{w}_k^m - \mathbf{w}_{k-1}^m = \boldsymbol{\theta}_{k-1}^+ - \boldsymbol{\theta}_{k-1}^- \ \forall k = 2...K \\
& \boldsymbol{\theta}_k^+ \geq 0, \boldsymbol{\theta}_k^- \geq 0 \ \forall k = 1...K-1 \\
& -t \leq \mathbf{1}^T (\boldsymbol{\theta}^+ + \boldsymbol{\theta}^-) \leq t \\
& t \geq 0
\end{aligned} \tag{7}
$$

where $\boldsymbol{\theta}^+, \boldsymbol{\theta}^- \in R^{K-1}$ and t is a scalar. (7) is a small-scale QP and hence can be solved by standard convex solver such as cvx, which is guaranteed to find the optimum of (6).

In summary, the whole learning algorithm is to run Algorithm 1 in a sufficient number of outer iterations until convergence, and in each outer iteration, solve M subproblems (6) by solving problem (7). We call our method DTRTS (Dynamic Trajectory Regression regularized by Temporal Smoothness) for subsequent reference.

Experiments

We conduct experiments to evaluate our framework on both synthetic and real-world trajectory data sets. The synthetic experiment, where the ground truth is provided, aims to demonstrate the high accuracy of our model in uncovering latent temporal dynamics of road costs, while the experiments on real-world data show the existence of smoothly changing road costs in real traffic, and that the accuracy of travel time prediction can be improved when such smoothness is explicitly taken into account.

Experiments on Synthetic Data

The goal of the synthetic experiment is to compare the temporal dynamics of road segments uncovered by our framework with the ground-truth to evaluate the effectiveness of our method. To that end, we firstly generate temporally changing costs for each link in a real-world road network in a manner that encourages both spatial and temporal smoothness, and then generate a set of trajectories and decide their travel time using those costs with random noises added. In particular, we use a region in Beijing as our road network containing 1501 nodes and 2372 edges, and then perform the following steps. First, we divide the time domain (6:00-23:00) into 34 equal time slots with each being half an hour and generate a set of smooth temporal dynamics typical in real world such as *"[slot 1-3: 60; slot 4-7: 30; slot 8-22: 50; slot 23-26: 20; slot 27-34: 60]"* (congested in both early and late rush hours) and *"[slot 1-34: 30]"* (strict speed limit without temporal variation), where the values here (say, *60*) mean speed (km/h), which can be seen as the reciprocal of unit cost. Second, we select 4-6 subregions as seed regions, and for each of them, assign a cost temporal dynamics randomly picked up to each link in that seed region. Gaussian noises are added to the cost in each time slot before each link-specific assignment is done in a seed region. Third, we propagate the cost temporal dynamics from seed regions to other unassigned road segments in a spatially and temporally smooth manner, that is, each time when a link propagates its cost temporal dynamics to one of its neighbors, it slightly adjusts the length of each smooth interval by moving the interval adjacency positions a bit; also, it adjusts the mean value of each time interval (the speed value in the previous example) by adding a Gaussian noise and then perturbing the cost in each time slot around the new mean. The final cost temporal dynamics for each link is obtained by computing the weighted average of all temporal dynamics that particular link receives from its neighbors with the weight inversely proportional to its hops from the seed region generating that temporal dynamics.

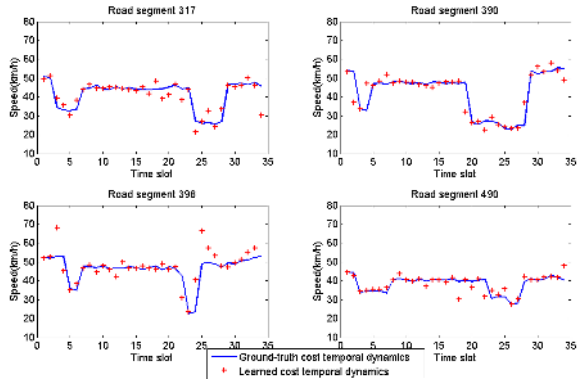

Figure 1: Cost temporal dynamics fitting results of some selected roads

Next, to generate a set of trajectories, we partition the region into four grids. To generate one trajectory, we randomly pick up two nodes from two randomly picked grids as source and destination and then generate a reasonable path connecting them. We then pick up a time point as the starting time point for this trajectory, and estimate the travel time it has spent on passing each link by using the pre-determined road temporal dynamics with Gaussian noise added. After that, we partition each trajectory into several subtrajectories such that each of them falls into one single time slot and estimate its total duration. In the last, we group all subtrajectories in the same time slot together, forming 34 sets of trajectories, where each trajectory is recorded as a sequence of road segment ids labeled only with its total travel time.

We then apply our framework to see if it can uncover the latent temporal dynamics for link using the synthetic data. The regularization parameters λ, α and β are tuned by cross validation, and their impact on model performance is omitted due to page limit. We evaluate both the consistency with the ground-truth and the smoothness of the learned temporal dynamics. The metric for consistency evaluation is root mean square error (RMSE), and the metric for smoothness evaluation is the mean (average) absolute value of successive cost differences (MASD), which is used for the temporal regularization in (5). We firstly visualize the learning results for 4 selected links in Figure 1, where the blue solid curves are the ground truth and the red crosses are the learned ones. It shows that the temporal dynamics learned has well captured the local smooth profiles of the underlying temporal dynamics. We then plot the link count distribution for each metric in Figure 2, which clearly shows that the learning results for most links achieve high consistency with the ground truth and good local smoothness, as evidenced by their small RMSE and MASD values.

Experiments on Real Data

For the experiment on real data, we show the existence of smoothly changing cost temporal dynamics in city traffic and demonstrate that the performance of trajectory travel time prediction can be greatly improved by explicitly capturing such temporal characteristics in the model.

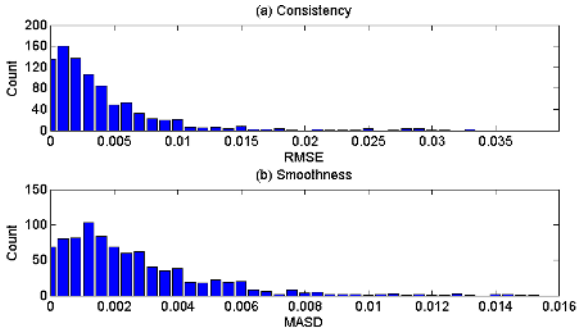

Figure 2: Consistency and smoothness of the learning results

For our experiment, we use a taxi trajectory data set collected from Shanghai, China, and select one week's data containing about 16500 trajectories covering 1000 taxis. We select a downtown area in Shanghai with an area of $108km^2$, and only retain trajectories that have passed that region. We then perform the same preprocessing steps as done in the synthetic experiment including dividing time slots and splitting trajectories to prepare the input to our framework.

In the absence of ground-truth, we evaluate the accuracy of travel time prediction for test trajectories using the temporal dynamics learned by performing 5-fold cross validation on the time-partitioned trajectory data set. In particular, we compare the prediction accuracy of our method DTRTS with two baselines: STR (Static Trajectory Regression) and NDTR (Naive Dynamic Trajectory Regression). STR ignores the presence of temporal dynamics in road segment costs and simply learns a static cost for each road segment, which is a degeneration of our method to the framework in (Idé and Sugiyama 2011). At another extreme, by setting λ to 0 in (5), NDTR separates the time domain into 34 time slots and then learns link costs in each time slot using only the trajectories falling in that slot, which has ignored the correlations between travel costs in neighboring time slots. The comparison of our method with these two baselines aims to show that to improve the accuracy of travel time prediction, it is crucial to capture the temporal variation of travel costs in real-world road networks while at the same time encouraging temporal smoothness to overcome sparsity problem.

To empirically justify the temporal sparsity inherent in the data set, we select one day and two small regions and show the number of passed trajectories in each time slot in Figure 3. The insufficiency of trajectories in many time slots does not mean that few vehicles passed the regions in those time slots, but that the passed taxis recorded are rare. This sparsity makes it essential to exploit trajectories in neighboring time slots in a principled way to encourage temporal smoothness.

To illustrate the results, we randomly pick 25% of trajectories from the test set and plot the comparison between their recorded and predicted travel time in Figure 4, where the 45° line represents a perfect agreement. The Pearson correlation coefficients of DTRTS, STR and DTR are 0.9057, 0.7363 and 0.8557, respectively. We can see that our method, which tracks temporal dynamics while at the same time encourag-

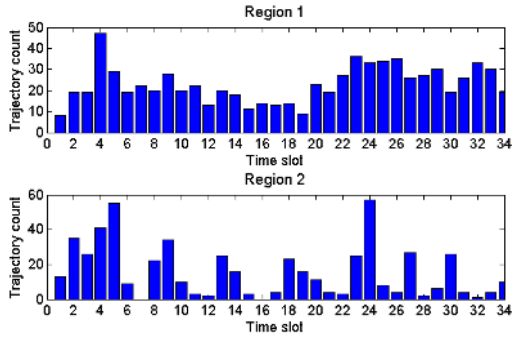

Figure 3: Empirical evidence for temporal sparsity

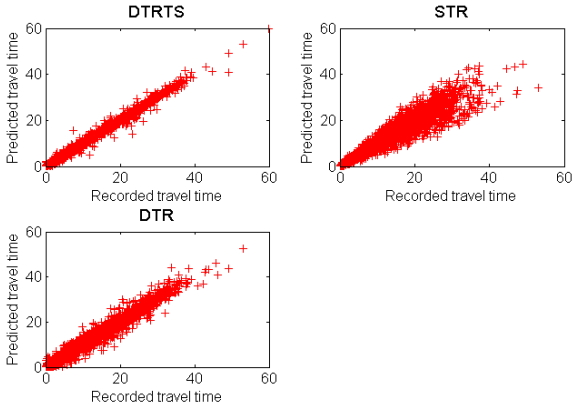

Figure 4: Comparison of 3 methods on travel time prediction

ing temporal smoothness, clearly outperforms the other two. The performance of STR degrades as the trajectory length increases. This is because long trajectories normally span multiple time slots but STR is unaware of temporal dynamics. DTR shows a poorer generalization performance than DTRTS because it is trying to fit too many parameters without imposing effective temporal regularization. Next, we compare three methods in how their prediction RMSE (in *minutes*) on test set change as the training proceeds in Figure 5. The result clearly shows that DTRTS gives the best performance. STR converges very soon when prediction performance is still poor, partly because the region we selected for experiment is a prosperous downtown region, whose traffic flow shows high variation over time.

Related Work

Traffic mining-based fastest route computation Many approaches have been proposed to recommend fastest route by mining knowledge from historical vehicle trajectories. For example, (Gonzalez et al. 2007) derived inductive rules from augmented historical traffic database describing how various contexts decide road speed patterns. This approach is infeasible in most cases as such a database is hard to obtain in practice. (Idé and Sugiyama 2011) formulated the task of learning road travel costs from trajectory data sets as a ridge regression problem with the regularization term

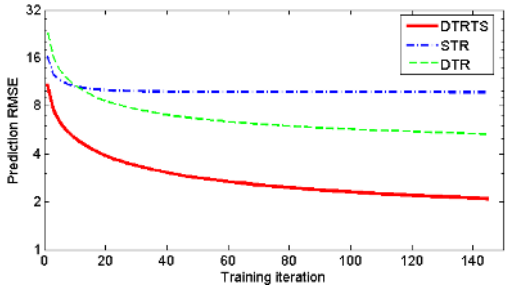

Figure 5: Comparison of 3 methods on the convergence of their travel time prediction performance

encouraging spatial smoothness. However, its ignorance of the presence of temporal variation of road travel costs in reality makes it somewhat impractical. (Yuan et al. 2010) built a time-dependent landmark graph by mining from historical traffic data for fast route computation. However, it lacks effective mechanism to propagate the learned knowledge along spatial or temporal dimension, making it vulnerable to spatial or temporal sparsity that might occur in real-world data sets. Our work aims to overcome these limitations.

Shortest path search on time-dependent graphs With temporal variation considered, fastest route computation is modeled as a time-dependent shortest path search problem. While numerous works (Kanoulas et al. 2006; Orda and Rom 1990; Nachtigall 1995) have proposed efficient algorithms to tackle this problem, most of them unrealistically used prior knowledge of the temporal functions of edge weights. Our work aims to compensate this drawback by learning the temporal dynamics from historical traffic data.

Multi-task learning Multi-task learning aims to overcome data sparsity problems in multiple domains by allowing information sharing among each other. The key step is to model the commonality among different tasks in an appropriate way, which is highly application-dependent. There are theoretical works that capture the commonality in model parameters (Evgeniou and Pontil 2004), feature representation (Argyriou, Evgeniou, and Pontil 2008), and task relationships (Zhang and Yeung 2010). Our work treats each time slot as a separate domain and aims to capture the commonality linking multiple model parameters. Such commonality is local temporal smoothness in our scenario, which has not been modeled before. For this reason, we propose our own method to model such specific commonality.

Lasso-based penalized techniques Encouraging temporal smoothness amounts to encouraging sparsity of successive parameter differences. In statistics, lasso (Fu 1998) is a technique widely used to encourage sparsity in parameter estimation. A variation of lasso called fused lasso (Tibshirani et al. 2005) was proposed to encourage sparsity of successive parameter differences, which clearly meets our demands in modeling local temporal smoothness. However, their solutions cannot be directly used as multiple fused lassos are coupled by regression square loss in our objective. Hence we resort to block coordinate descent method (Tseng 2001) to solve our problem effectively and efficiently.

Conclusion and Future Work

In this paper, we propose a multi-task framework to learn temporal dynamics of road travel costs from historical trajectory data, which can serve many smart city applications. To capture the periodic temporal variation in travel costs, we extend a static trajectory regression framework to a multi-task one by regarding each fine-grained time slot as a separate task. To overcome the temporal sparsity problem, we introduce a novel cross-task regularization motivated by fused-lasso to encourage temporal smoothness, based on the observation that the traffic flow usually changes smoothly over time. We then propose an efficient block coordinate descent algorithm to solve the optimization problem and prove its convergence to global optimum. Experiments on synthetic data show our model's ability in effectively uncovering the cost temporal dynamics and experiments on real data demonstrate our model's good generalization performance. For future work, it would be interesting to extend the framework to detect abnormal traffic events, capture traffic variation between different time periods(e.g, weekdays, weekends), and take various contexts (e.g, weather, holidays) into consideration.

Acknowledgements

This research was supported by Hong Kong, Macao and Taiwan Science & Technology Cooperation Program of China under Grant No. 2012DFH10010, Science and Technology Planning Project of Guangzhou China under Grant No. 2012Y2-00030, and Huawei Corp. Contract YBCB2009041-27.

Appendix: Proof of Lemmas/Theorems

Proof of Lemma 1. We first show that the sublevel set $S = \{\mathbf{W}|f(\mathbf{W}) \leq f(\mathbf{W}^0)\}$ is bounded. Assume on the contrary that it is not bounded, then, there must exist a unit direction $\mathbf{V} \in R^{M\times K}(||\mathbf{V}||_F = 1)$ such that $\forall\lambda > 0$, $\lambda\mathbf{V} \in S$, i.e., $\forall\lambda > 0$, $f(\lambda\mathbf{V}) \leq f(\mathbf{W}^0)$. However, the expression of $f(\lambda\mathbf{V})$ has a term $\sum_{k=1}^{K}\lambda^2\mathbf{V}_{\star k}^T\mathbf{V}_{\star k} = \lambda^2||\mathbf{V}||_F(\mathbf{V}_{\star k}$ denotes the kth column of $\mathbf{V}$ and $||\mathbf{V}||_F$ denotes the Frobenius norm of $\mathbf{V}$). Since $||\mathbf{V}||_F = 1 > 0$, we must have $\lambda^2||\mathbf{V}||_F > f(\mathbf{W}^0)$ provided $\lambda > \sqrt{f(\mathbf{W}^0)}$. In addition, all other terms in $f(\lambda\mathbf{V})$ are non-negative by their construction, so $f(\lambda\mathbf{V}) > f(\mathbf{W}^0)$ when $\lambda > \sqrt{f(\mathbf{W}^0)}$, resulting in an contradiction. Therefore S is bounded.

Then we show that S is closed. According to the definition of closed set, we only need to prove that $S' = \backslash S = \{\mathbf{W}|f(\mathbf{W}) > f(\mathbf{W}^0)\}$ is open. Assume on the contrary that S' is not open, then, there exists a point $\mathbf{X} \in S'$ which is not an interior point, that is, there exists a unit direction $\mathbf{V}$ such that $\forall\delta > 0$, $\mathbf{X} + \delta\mathbf{V} \notin S'$. Or equivalently, (i)$f(\mathbf{X}) > f(\mathbf{W}^0)$, and (ii)$\forall\delta > 0$, $f(\mathbf{X} + \delta\mathbf{V}) \leq f(\mathbf{W}^0)$. But this is impossible because, suppose $f(\mathbf{X}) = \theta > f(\mathbf{W}^0)$, due to the continuity of $f(\mathbf{W})$, we must have that $\lim_{\delta\to 0} f(\mathbf{X} + \delta\mathbf{V}) = f(\mathbf{X}) = \theta > f(\mathbf{W}^0)$, which contradicts (ii). Therefore, S is closed. □

Proof of Lemma 2. $f = f_0 + \lambda\sum_{m=1}^{M} f_m$. f_0 is a quadratic form of $\mathbf{W}$, and by the positivedefiniteness of $\mathbf{\Phi}_k^T\mathbf{\Phi}_k$ and Laplacial matrix $\mathcal{L}$, it is easy to see that f_0 is convex on $\mathbf{W}$. Next we consider $\sum_{m=1}^{M} f_m$. We introduce matrix $\mathbf{A} \in R^{K\times(K-1)}$ where $\mathbf{A}_{kk} = -1$ and $\mathbf{A}_{k+1,k} = 1$ for all $k = 1...K-1$, and all other elements are 0. Also denote $\mathbf{e}_m \in R^M$ as a basis vector with the mth element being 1 and all other elements being 0. Then, $\sum_{m=1}^{M} f_m$ can in fact be rewritten as $\sum_{m=1}^{M}||(\mathbf{WA})^T\mathbf{e}_m||_1^2$. Note that $||(\mathbf{WA})^T\mathbf{e}_m||_1$ is convex on $\mathbf{W}$ because it is the l_1 norm(convex function) of an affine transformation of $\mathbf{W}$ according to (Boyd and Vandenberghe 2004). Also we have $||(\mathbf{WA})^T\mathbf{e}_m||_1$ is non-negative. Then, by the conclusion in Example 3.14(page 87)(Boyd and Vandenberghe 2004), $\sum_{m=1}^{M}||(\mathbf{WA})^T\mathbf{e}_m||_1^2$ is a convex function on $\mathbf{W}$. Therefore, f is the non-negative weighted sum of two convex functions, and hence is convex on $\mathbf{W}$ according to (Boyd and Vandenberghe 2004) □

Proof of Theorem 1. By Lemma 1, $S = \{\mathbf{W}|f(\mathbf{W}) \leq f(\mathbf{W}^0)\}$ is closed (1). Obviously, f is continuous in domain S (2). By Lemma 2, we know that f is convex on $\mathbf{W}$, hence f is convex(also puedoconvex) in $(\mathbf{w}^m, \mathbf{w}^{m'})$ for every $m, m' \in \{1, 2, ..., M\}$, that is, f is puedoconvex on every pair of blocks of $\mathbf{W}$ where the blocks are partitioned by rows of $\mathbf{W}$ as before (3). The domain of f_0 is $R^{M\times K}$, which is obviously open. Also, f_0 is differentiable on its domain (5). Then, by Lemma 3.1 in (Tseng 2001), we have that f is regular at every $\mathbf{W} \in S$, where the regularity is defined in (Tseng 2001) (6). Combining (1), (2), (3), (6), and making use of Theorem 4.1(a) in (Tseng 2001), we conclude that the $\mathbf{W}$ sequence generated by the block coordinate descent method in algorithm 1 will converge to a stationary point of f. Also by Lemma 2, the only stationary point of f on its domain is its global minimum, so we conclude that algorithm 1 will converge to the global minimum of f. □

References

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2008. Convex multi-task feature learning. *Machine Learning* 73(3):243–272.

Auslender, A. 1976. *Optimisation: méthodes numériques*. Masson.

Bertsekas, D. P. 1999. Nonlinear programming.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Evgeniou, T., and Pontil, M. 2004. Regularized multi–task learning. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Dining, KDD'04*, 109–117. ACM.

Fu, W. J. 1998. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics* 7(3):397–416.

Gonzalez, H.; Han, J.; Li, X.; Myslinska, M.; and Sondag, J. P. 2007. Adaptive fastest path computation on a road network: a traffic mining approach. In *Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB'07*, 794–805.

Idé, T., and Kato, S. 2009. Travel-time prediction using gaussian process regression: A trajectory-based approach. In *Proceedings of the 9th SIAM International Conference on Data Mining, SDM'09*, 1183–1194.

Idé, T., and Sugiyama, M. 2011. Trajectory regression on road networks. In *Proceedings of the 25th AAAI Conf. on Artificial Intelligence, AAAI'11*, 203–208.

Kanoulas, E.; Du, Y.; Xia, T.; and Zhang, D. 2006. Finding fastest paths on a road network with speed patterns. In *Proceedings of the 22nd International Conference on Data Engineering, ICDE'06*. IEEE.

Nachtigall, K. 1995. Time depending shortest-path problems with applications to railway networks. *European Journal of Operational Research* 83(1):154–166.

Orda, A., and Rom, R. 1990. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM (JACM)* 37(3):607–625.

Tibshirani, R.; Saunders, M.; Rosset, S.; Zhu, J.; and Knight, K. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(1):91–108.

Tseng, P. 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications* 109(3):475–494.

Yuan, J.; Zheng, Y.; Zhang, C.; Xie, W.; Xie, X.; Sun, G.; and Huang, Y. 2010. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 99–108. ACM.

Zhang, Y., and Yeung, D. 2010. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, UAI'10*, 733–742.