Temporal Modeling of Spatiotemporal Networks

Ugur Demiryurek, Bei Pan, Farnoush Banaei-Kashani and Cyrus Shahabi
Department of Computer Science
University of Southern California
Los Angeles, CA 90089
{demiryur,beipan,banaeika,shahabi}@usc.edu

ABSTRACT

A spatiotemporal network is a spatial network (e.g., road network) along with the corresponding time-dependent travel-time for each segment of the network. Design and analysis of policies and plans on spatiotemporal networks (e.g., for path planning with location-based services) require realistic models that accurately represent the temporal behavior of such networks. In this paper, for the first time we propose a temporal modeling framework for spatiotemporal networks that enables 1) generating an accurate temporal model from real temporal data collected from any spatiotemporal network (so as to be able to publish the temporal model of the spatiotemporal network without having to release the real data), and 2) augmenting any given spatial network model with a corresponding realistic temporal model custom-built for the specific spatial network (in order to be able to generate a spatiotemporal network model from a solely spatial network model). We validate the accuracy of our proposed modeling framework via experiments. We also use the proposed framework to generate the temporal model of the Los Angeles County freeway network and publish it for public use.

1. INTRODUCTION

The latest developments in online map services (e.g., Google Maps) and their widespread usage in hand-held devices and car-navigation systems have led to the recent prevalence of the location-based services. Many of the location-based services rely on efficient computation of the travel-time between a source and a destination in a spatial network. While the majority of the previous studies (e.g., [16, 10, 12, 3]) simplistically assume the travel-time of each segment of the network is constant, in reality the actual travel-time of a segment heavily depends on the traffic flow on the segment; hence, a variable function of time. Recently, an increasing number of new studies [8, 4, 5] consider travel-time computation in *spatiotemporal networks*, i.e., spatial networks along with the corresponding time-dependent travel-time for each segment of the network. However, most of these studies resort to using simplistic models and/or synthetic datasets to represent the temporal aspect of the spatiotemporal networks, mainly because collecting and working with real temporal data from spatial networks is costly and difficult, and the available temporal datasets are often proprietary and cannot be released for public use. Obviously, inaccurate temporal representation of spatiotemporal networks can seriously affect the validity of the design and evaluation of any proposed path planning techniques for such networks; hence, the need for realistic models for traffic flows in spatiotemporal networks.

In this paper, we propose a framework for realistic and accurate modeling of traffic flows in spatiotemporal networks. The benefit of the proposed framework is twofold. First, anyone (e.g., governmental agencies) in possession of a real temporal dataset collected from a spatiotemporal network can use the proposed framework to derive and generate a realistic temporal model for the corresponding network, to be shared for public use (e.g., released to researchers and policy planners) without infringing the laws of possession and jeopardizing the privacy of the dataset. As an example, we have used the proposed framework to generate and publish a realistic model for traffic flows in all freeways of the Los Angeles County based on the real (and proprietary) data provided to us by the county (see Section 4.1 for more details about this dataset)[1]. Second, as we describe in Section 4 (since the traffic in Los Angeles County is arguably typical and generic) one can use the proposed framework to generate realistic temporal data specific to and customized for any given spatial network; hence, augmenting the spatial network model to the corresponding spatiotemporal network model. In the second case, we use a semi-supervised hierarchial clustering approach (based on the spatial characteristics of the network) to generate the spatiotemporal model of the network. To the best of our knowledge, our work is the first attempt in generating realistic temporal models for spatiotemporal networks.

The remainder of this paper is organized as follows. In Section 2 we review the related work. In Section 3 we provide the preliminary definitions, and subsequently in Section 4 we establish the theoretical foundation of our proposed traffic flow modeling framework and discuss the three-phase modeling process of this framework. In Section 5, we present the results of our experiments to verify and validate the accuracy of this framework. Finally, in Section 6 we conclude and discuss our future work.

2. RELATED WORK

In [2], Brinkhoff et al. introduces a system called Network-based Generator of Moving Objects that models and simulates the behavior of the moving objects (e.g., vehicles) on spatial networks. This system has been extensively used to benchmark k-nearest neighbor and location based search algorithms in spatial networks. While the focus of this system is the moving objects and their mobility in spatial networks, we primarily study to model the traffic flow of

IWCTS '09 November 3, 2009, Seattle, WA, USA.
Copyright ©2009 ACM ISBN 978-1-60558-861-2 ...$10.00.

[1] http://geodb.usc.edu:8080/transdec/model.html

the network segments. In addition, this work relies on some simplistic assumptions about the network parameters such as minimum and maximum speed assignment for the segments, and number of moving objects in the system.

The freeway Performance Evaluation Monitoring System (PeMS) [13] developed by UC Berkeley collects and stores data from loop detectors operated by Caltrans. The main goal of PeMS is to convert the freeway sensor data into graphs and tables that show performance measures and traffic patterns on freeways in the State of California. The scope of PeMS is limited to collection and analysis of the historical freeway sensor data. However, our goal is to model the traffic flow for any given spatial network (even without sensor data) as described in Section 4.

Most of the traffic simulators developed in the recent decade use microscopic simulation models (aka, agent-based models) [14, 6] to simulate the traffic flow in spatial networks. The microscopic simulation models focus on the behavior of the system entities (e.g., vehicles and drivers) as well as their interactions with the system parameters (e.g., traffic lights). For instance, for each vehicle in the stream, a lane-change is described as a detailed chain of drivers' decisions. These simulation models, however, ignore the global descriptions of the traffic flows such as flow-rate, density and velocity and often are restricted to synthetic or simplified data.

There also exist several machine learning techniques developed for the purpose of traffic modeling. In [7], Kamarianakis et al. proposed a space-time autoregressive integrated moving average model to estimate the traffic flows on spatial networks. In [9], Lint et al. introduced a neural network based technique to model the traffic flow on freeways. However, all of these approaches are univariate and ignore most important factors such as road network geometry and spatiotemporal characteristics of the traffic flow.

3. DEFINITIONS

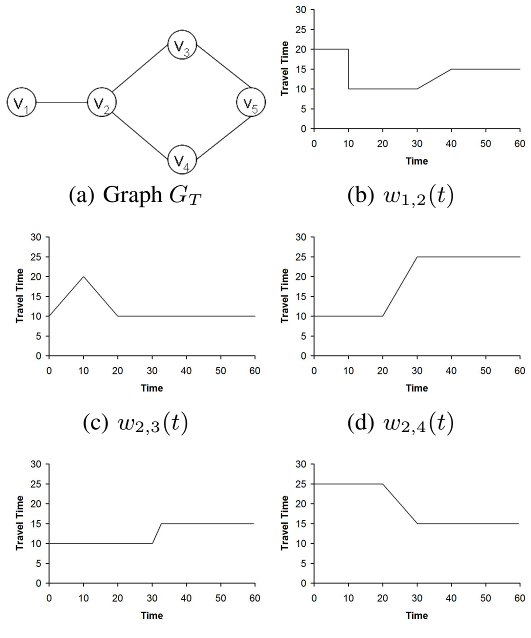
In this section, we formally define spatiotemporal network. We assume a spatial network (e.g. the Los Angles road network) containing a set of nodes and segments. We model the spatial network as a time-dependent weighted graph (i.e., spatiotemporal network) where the weights are time-varying travel-times (i.e., traffic flow) between the nodes. Below, we formally define our terminology

DEFINITION 1. *Spatiotemporal Network*
A Spatiotemporal Network is defined as a graph $G_T(V, E, W)$ where $V = \{v_i\}$ is a set of nodes representing the intersections and terminal points, and E ($E \subseteq V \times V$) is a set of edges representing the network segments each connecting two nodes. Each edge e is represented by $e(v_i, v_j)$ where v_i and v_j are starting and ending nodes, respectively, and $v_i \neq v_j$. For every edge $e(v_i, v_j) \in E$, there is an edge travel-time function $w_{i,j}(t) \in W$, where t is the time variable in time domain T. An edge travel-time function $w_{i,j}(t)$ specifies how much time it takes to travel from v_i to v_j starting at time t.

Figure 1 illustrates a spatiotemporal network modeled as $G_T(V, E, W)$. While Figure 1(a) shows the network structure with five nodes and five edges, Figures 1(b), 1(c), 1(d), 1(e), 1(f) illustrate the time-dependent edge costs (i.e., travel-times) for the edges of the network.

4. METHODOLOGY

Our modeling framework is based on the real-world traffic data collected from the freeways in Los Angeles County (LA). The proposed framework offers solutions to the following two cases. In the first case, given the historical temporal data (time-series of traffic flow possibly collected from various sensor locations) of a spatial network, our framework creates the spatiotemporal model of that network using the temporal data only. We refer to this case as Modeling with Temporal Data (MTD). However, the temporal data may not be available for most of the spatial networks as acquiring such data is a complex and sometimes prohibitively expensive task. In this (second) case, our framework generates a spatiotemporal network model from the *spatial characteristics* and the topology of the spatial network. We refer to the second case as Modeling with Spatial Characteristics (MSC).



(a) Graph G_T

(b) $w_{1,2}(t)$

(c) $w_{2,3}(t)$

(d) $w_{2,4}(t)$

(e) $w_{4,5}(t)$

(f) $w_{3,5}(t)$

Figure 1: A Spatiotemporal network $G_T(V, E, W)$

Our approach includes the following three steps. The first step is traffic flow generation where we compute the time-dependent travel-times on each network segment using historical sensor data. In the second step, we attach semantic information to the network by labeling the regions of the network based on its spatial characteristics. We refer to this step as *spatial characterization*. In the third step, we employ a semi-supervised clustering algorithm to group the traffic flows of similar kind into respective spatial characteristics. This step enables us to find the most representative traffic flows of the network regions based on their spatial characteristics. While the techniques developed in the first step can directly be used in MTD, we employ the second and third steps to address MSC. Below, we explain these steps in detail.

4.1 Traffic Flow Generation

In the past one year, through a system called RIITS [15], we have been continuously collecting and archiving the sensor (i.e., loop detector) data from a collection of approximately 1500 sensors located on the freeways of LA County. The urban area of Los Angeles County has an area of 4752 square miles (12,308 km^2) and population of approximately nine million people. Figure 2 shows the spatial span (covering 1183 miles) of the traffic sensors on a map. The sampling rate of the sensor data is 1 reading/sensor/min. We average the readings over three consecutive time intervals in order to ease the implementation and smooth out the noise. Therefore, each sensor provides 480 distinct measurements per day. We

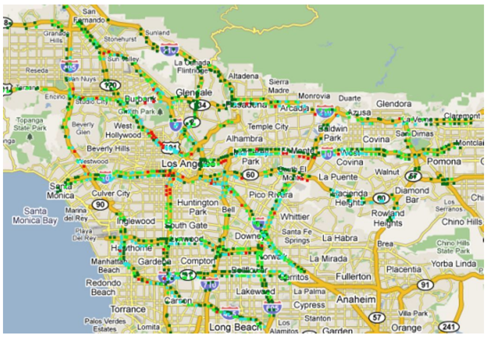


Figure 2: Traffic sensor layout in LA County

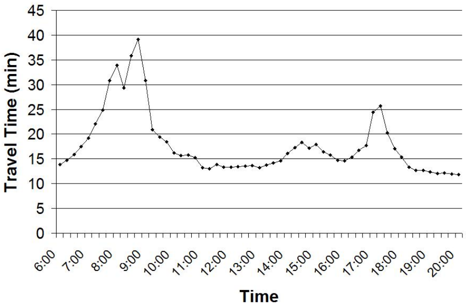


Figure 3: Real travel-time during a weekday on a segment of I-405 in LA County

only consider the readings from the weekdays. The storage space required for this streamed dataset is approximately 350 MB/day without indexing overheads. Currently, our data warehouse consists of data from the period of October 2008 to June 2009.

The main traffic parameters collected from the loop detectors are *occupancy* and *volume*. The loop detectors turn on and off as cars pass over them. The number of *'on'* readings within a time interval (e.g., 60 seconds) determines the occupancy measure. Occupancy is defined as the percentage of time a point on network segment is occupied by vehicles. The other parameter, volume, is defined as the number of vehicles flowing past a point during a time interval. We derive the third parameter, i.e., speed, from the occupancy and volume readings using the formula introduced in [1] $Speed = \frac{C*V}{O}$ where C is a constant proportional to the average length of a car, V is volume, and O is occupancy.

In order to determine the time-dependent travel-time on each network segment, we employ a two step process. First, using the spatial query operators, we map the coordinates of the individual sensors to network segments. Then, for each segment, we aggregate the desired traffic measure in both time and space dimensions by considering the distances between the sensors. For instance, for a given time instance (i.e., t), we compute the travel-time of a segment by the following formula $Travel - Time = \sum_{i=1}^{n} \frac{D(s_i, s_{i+1})}{S_i}$ where S_i, $D(s_i, s_{i+1})$ and n represents the speed measured on sensor i at time t, distance between two consecutive sensors, and number of sensors on the segment, respectively. To illustrate, consider Figure 3 that shows the graph of travel-time (aggregated for each 15 minutes) on a segment of I-405 freeway in LA between 6:00 AM and 8:00 PM on a weekday.

4.2 Spatial Characterization

In this section, we describe how we characterize the spatial network using geographical and topological characteristics of the network. Studying the real-world traffic data, we observe the following three main spatial and temporal characteristics of the traffic flow which motivated us to pursue the approach discussed in Section 4.3. First, the traffic flow on network segments demonstrates a strong *periodicity* at various spatial and temporal scales (daily, weekly, monthly, and quarterly). For example, the traffic flow on particular segment may exhibit a huge peak on each day at around 8:00 AM, a smaller one at around 4:00 PM, and an absolute minimum at around 2:00 AM during the weekdays in fall season. Second, the traffic flow is highly affected by the *spatial characteristics* of the network. That is, the traffic flow follows different patterns near major residential areas, city centers (aka, downtown), attraction areas (e.g., shopping centers, sports stadiums), and in regions between them. For instance, while a segment connecting a residential area to downtown is congested during morning hours, another segment connecting downtown to a residential area is usually congested in the afternoon. Third, the traffic flows are also affected by the topology (i.e., another spatial characteristic) of the network. For example, a dense network topology which contains numerous nodes (hence many alternative routes) is usually congested in the hubs (i.e., intersection of the nodes) depending on the time of the day but has steady traffic flow in the rest of the region.

As we discussed, the main idea behind incorporating the spatial characteristic of the network to our model comes from the observation that the traffic flow in certain parts of the network can be affected by the geographical and topological characteristics. Although, there are various other characteristics (e.g., population and demographics) that are likely to be offered to characterize a spatial network, we select two major characteristics for the purpose of this study namely, geographical region and density. We plan to include more spatial characteristics into our model in the future. With our study, we developed a graphical user interface (i.e., a map mashup) that enables users to label the geographical regions (i.e., residential, downtown, and attraction) of the spatial network. To capture the density information, the map interface allows users to partition the spatial network into regular grid cells (e.g., 5x5 km) and label the sub-networks (overlapping the grid cells) as dense or sparse based on the distribution of the number of nodes in each grid cell. Note that the map interface allows users to control the grid cell size. Clearly, these characteristics do not consider all possible aspects of the traffic flow and their specific definitions may vary. Our main focus is to establish a framework that considers the spatial characteristics of the network for generating a spatiotemporal network model. We emphasize that our framework allow users to select their preferred spatial characteristics among the pre-defined ones. For example, with our case, one can only select regional information (ignoring density) to generate the spatiotemporal model of a particular network. In the following section, we explain how we incorporate the spatial characteristics of a network in to our proposed semi-supervised clustering algorithm.

4.3 Hierarchical Semantic Traffic Flow Clustering

In this section, we explain our proposed Hierarchical Semantic Traffic Flow Clustering ($HSTFC$) method that is based on the semi-supervised clustering algorithm proposed in [17]. Although the unsupervised clusters can identify the natural groups, it is extremely difficult to construct the mapping between the representation of the groups and their semantic meanings. Semi-supervised clustering addresses this issue by relating prior knowledge (in the form of labels and constraints) in to clusters. In other words, semi-supervised clustering not only creates natural groups with similar features but also provides semantic meanings to the cluster results. Therefore, in the context of our problem, semi-supervised clustering technique enables us to associate spatial characterization (i.e., semantic information) of the network with the traffic flows.

In the following sections, we first explain pairwise constraint clustering (a semi-supervised clustering method) and discuss how it fits in to our problem. Second, we present our proposed hierarchical clustering structure.

4.3.1 *Pairwise Constraint Clustering Method*

Pairwise constraint clustering (PCC) [17] is a classic technique to employ semi-supervised clustering. PCC introduces prior knowledge in the form of pairwise constraints. In particular, PCC incorporates the pairwise *cannot-link* and *must-link* constraints of the data instances, and make the cluster results maximally satisfy all the constraints. While *must-link* constraint specifies that two instances should be assigned into one cluster, *cannot-link* constraint specifies that two instances should be assigned into different clusters. Let us now explain how this technique is adopted to our problem. As we discussed, in typical transportation networks, segments demonstrate different traffic patterns based on their geographical areas. For example, the traffic pattern of freeways near downtown may be entirely different than that of a suburban area. On the other hand, the segments which are spatially close to each other (e.g., two freeway segments near Hollywood) may generate similar traffic patterns. With this example, we can consider applying the knowledge in the later case in the form of *must-link* constraint and the former case in the form of *cannot-link*. The formulation of pairwise constraint clustering is given below.

Let M be the set of must-link pairs such that $(x_i, x_j) \in M$ implies x_i and x_j should be assigned to the same cluster, and C be the set of cannot-link pairs such that $(x_i, x_j) \in C$ implies x_i and x_j should be assigned to different clusters. Let $W_m = w_{ij}$ and $W_c = \overline{w}_{ij}$ be the two sets that give the weight to the constraints in M and C respectively. Let l_i be the assigned cluster number of instance x_i, and μ_{l_i} be the centroid of the cluster l_i. The cost of violating these pairwise constraints is typically the sum of violating pair(s) times their penalty weight. Specifically, the cost of violating a must-link constraint is given by $w_{ij} * f(l_i \neq l_j)$, where f is the indicator function, with $f(true)$ = 1 and $f(false)$ = 0. Similarly, we could get the cost of violating the cannot-link constraint as $w_{ij} * f(l_i = l_j)$. Using this model, the problem of PCC is formulated as the minimization problem on the following objective function:

$$\frac{1}{2} \sum_{x_i \in D} \|x_i - \mu_{l_i}\|^2 + \sum_{(x_i, x_j) \in M} w_{ij} * f(l_i \neq l_j)$$
$$+ \sum_{(x_i, x_j) \in C} \overline{w_{ij}} * f(l_i = l_j)$$

Algorithm 1 presents our pairwise constraint (k-means) clustering algorithm. The algorithm takes the dataset of the traffic flow(D) , a set of must-link constraints (M), and a set of cannot-link constraints (C). Note that M and C is derived from the spatial characterization step. At first, we call the function *POPULATE-CONSTRAINTS* to generate transitive closure over pair-wise constraints denoted as M', C'. Then, we initialize the cluster center by choosing k points from the cannot-link constraints pairs in C' as long as they don't have must-link constraints in M'. If we cannot find such k points, we exit the algorithm to enrich the input constraint set from the dataset, and restart. Finally, the algorithm returns the centroid of clusters that satisfies all specified constraints. It is important to note that, with Algorithm 1, we utilize the pairwise constraints for initializing the cluster centroid. For example, if two instances have cannot-link constraint, they should have distinct spatial category information. This enables us to guide the clustering process that generates two clusters maintaining distinct spatial characterizations. We assume that the cluster number (i.e., k) is equal to the number of pre-defined categories.

Algorithm 1 Pairwise Constraint K-means Clustering Algorithm

```
Input: Traffic flow D, must-link constraints $M \subseteq D \times D$, cannot-link constraints $C \subseteq D \times D$, Cluster Number k
Output: The cluster index of each variable $l_1, ...l_n$
1: Call POPULATE-CONSTRAINTS($M, C$);
2: Initialize the cluster center $\mu_1, ...\mu_k$
3: For each point $x_i$ in $D$, assign it to the closest cluster $l_j$ such that VIOLATE-CONSTRAINTS($di, l_j, M, C$) is false.
4: For each cluster $C_i$, update its center by averaging all of the points $d_j$ that have been assigned to it.
5: Iterate between (3) and (4) until convergence.
6: Return $l_1, ...l_n$.
POPULATE-CONSTRAINTS(must-link constraints set M, cannot-link constraints set C)
1: For each a: if both $(a, b), (a, c) \in M$, $M = (b, c) \cup M$
2: For each a: if $(a, b) \in M$, and $(a, c) \in C$, $C = (b, c) \cup C$
3: Return $M, C$ and denoted as $M', C'$
VIOLATE-CONSTRAINTS(data point x, cluster L, must-link constraints M, cannot-link constraints C)
1: For each $(x, y) \in M$: If $y \notin L$, return true.
2: For each $(x, y) \in C$: If $y \in L$, return true.
3: Otherwise, return false.
```

4.3.2 *Hierarchical Clustering*

So far we have explained the pairwise constraint clustering, but PCC itself is not satisfactory for our problem. Since our ultimate goal is to find the representative curve for the network segments, and each specific segment has various spatial characterizations, we should apply different spatial characteristics in different levels. This is because, if we apply various spatial characteristics in one level, we may get contradictions among different characteristics. For example, let us consider both region and density feature as two types of characteristics that guide the clustering. During the must-link and cannot-link constraint construction, two instances which have the same density value may lead to a must-link constraint. However, a cannot-link constraint may be assigned to them because of their differences in the region values. In this case, due to the two instances have must-link and cannot-link constraints simultaneously, our clustering technique would have poor performance. To avoid this problem, we propose a hierarchical clustering method to arrange one type of characteristics to guide the clustering in one level. It is important to note that our hierarchical structure makes it very easy to add new characteristics (e.g., segment length) to the system. Currently, we only have two levels namely, region and density.

Fig.4 illustrates the schema of our hierarchical clustering method. With the dataset as the input, the first level applies the region feature to guide the clustering. In the second level, based on the cluster results from the first level, we utilize the density characterization to direct semi-supervised clustering. In the end, we have the centroid presentation of the category defined by the combination of two level spatial characteristics as output. Note that the order of the characterizations applied in the two levels are flexible to change.

Let us now explain how this step is useful to address MSC case discussed in Section 4. After the hierarchical clustering, we obtain the centroid for each semantic cluster based on the two type of characteristics (i.e. region and density). Therefore, these clusters could obtain meaningful labels constructed from the combination of the values from the two characteristics. In addition, we know

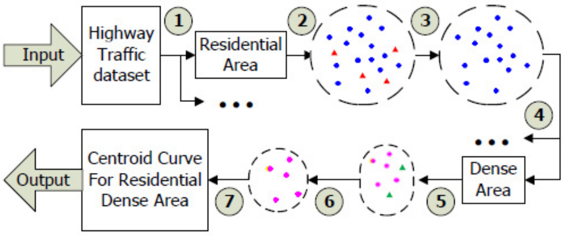


Figure 4: Hierarchical semantic clustering flowchart. (1) Clustering by region (2) First level cluster results (3) Result Refinement (4) Clustering by density (5) Second level cluster results (6) Result Refinement (7) Centroid computation

that the centroid of the clusters as the representative traffic flow for the area with corresponding label. Given a spatial network and its spatial characterization, we consider the values of the characteristics as the index to search for the corresponding cluster, and use the centroid of the result as the simulated traffic curve for the segments.

5. PERFORMANCE EVALUATION

5.1 Experimental Setup

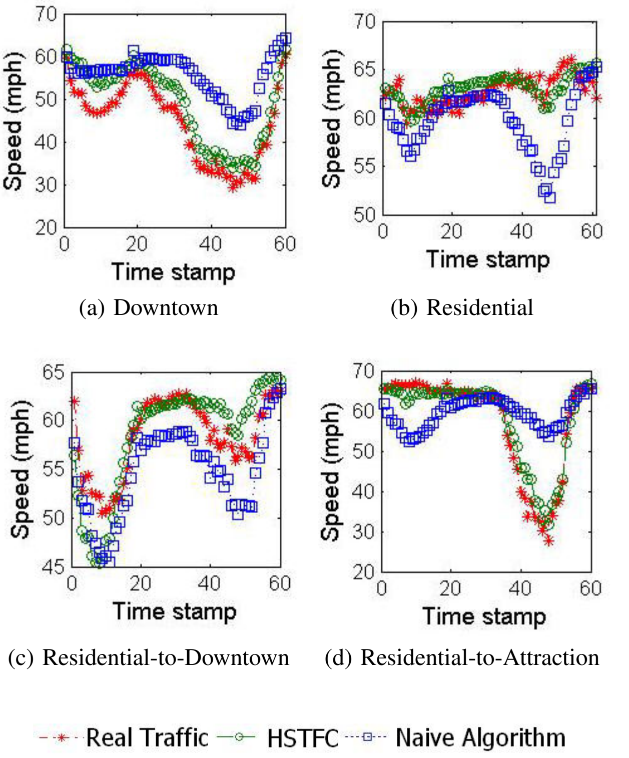
We conducted several experiments with different networks and parameters to evaluate the performance of our algorithm. As we mentioned in Section 4.1, we used real-world Los Angeles freeway traffic sensor data to construct our model. Since the traffic flow on freeways is much simpler than that of the local road network (i.e., no traffic light, no pedestrian), it requires less characterization. Therefore, to simplify our experiments, we only evaluate our model on freeway data. The sensor dataset is collected from 1592 sensors on the freeways during the period from October 2008 to June 2009. In order to represent the traffic flow on each segment, we compute the average travel time (from the historical sensor data) from 6:00 AM to 9:00 PM with 15 minute time intervals. As our spatial network dataset, we used Los Angeles (LA) and San Joaquin County (SJ) freeway network data. We obtained these datasets from NAVTEQ [11]. Using NAVTEQ dataset, we constructed the graph G(V,E) representation of LA and SJ freeway networks. Each network segment is represented in the vector data format and described by more than 20 attributes such as direction, speed limit, zip code, location, density, geographical location (e.g., residential) and etc. Based on the location and direction information, we labeled the freeway segments into eight spatial categories namely, *RR*, *R*, *D*, *A*, *R2D*, *D2R*, *R2A*, *A2R*. The descriptions of these labels are presented in Table 1. Moreover, in addition to region labels, we defined another label capturing the density information of the network segments. In order to assign density label to the network segments, we partitioned both LA and SJ freeway networks into 5 X 5 km regular grid cells. Based on the average number of nodes(α) in each grid cell (assuming uniform distribution of the nodes), we labeled the segments as *Dense Area* (i.e., area that has more nodes than α) or *Sparse Area*. We conducted our experiments on a workstation with 2.7 GHz Pentium Core Duo processor and 12GB RAM memory. Due to the space constraints, we only present the experimental evaluations from LA dataset.

5.2 Performance Study

For performance evaluation, we compared our algorithm with a naive approach that is based on decision tree technique. To implement decision tree, we used eight spatial categories (represented in Table 1) and density information (i.e., dense or sparse) as the nodes of the decision tree. The leaves of the decision tree contained the traffic flow information of the segments in the same category. Since each leaf can contain more than one traffic flow, we took the average value of the traffic flows in the corresponding leaves. This enabled us to represent each leaf with one traffic flow. With our experiments, we measured the traffic flow similarity, general error rate and confidence interval.

Table 1: Spatial Label Description

Label	Spatial Information for Freeway Segments
R	Residential Area
RR	Remote Area, area far from downtown and res.
D	Downtown Area
A	Attraction Area
R2D	From Residential Area to Downtown Area
D2R	From Downtown Area to Residential Area
R2A	From Residential Area to Attraction Area
A2R	From Attraction Area to Residential Area



(a) Downtown

(b) Residential

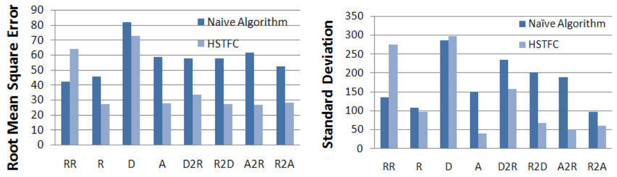
(c) Residential-to-Downtown

(d) Residential-to-Attraction

Figure 5: Instance comparison

5.2.1 Traffic Flow Similarity Comparison

With this experiment, we compare the traffic flow obtained from the two algorithms with actual (observed) traffic flow on the segments. We randomly choose one instance in four categories namely: downtown area, residential area, and residential-to-downtown area, residential-to-attraction area. Figure 5 shows the traffic flow with respect to these four categories. The traffic curves cover the period from 6:00 AM (represented as 0 in the figure) to 9:00 PM with 15 minutes time intervals. As illustrated, the traffic flow generated by our algorithm is more consistent with actual traffic flows. This is because, in real-world, some traffic patterns do not follow the major traffic flow trend in the same category due to some special events



(a) Mean (b) Variation

Figure 6: General error rate comparison

(e.g., accidents, lane closure). However, the naive approach considers that each instance contributes equally towards the construction of category presentation. This assumption causes the results deviate from the major pattern trend hence leading to imprecise traffic flow representation. On the other hand, HSTFC considers both the spatial correlations and the traffic flow feature; therefore the centroid is calculated only based on the major trend of each category without possible noisy instances.

5.2.2 General Error Rate Comparison

With the second set of experiments, we compare the overall performance of the two algorithms based on average root mean square error (MSE) and standard deviation(STD). These two techniques enable us to quantify the amount by which the estimated centroids differ from the real instances. The MSE and STD are calculated based on the distances between individual instance and its corresponding centroid. The lower the value of them, the more precise the corresponding algorithm is. Figure 6 depicts the performance of the two algorithms with respect to eight spatial categories. In general, the results show that the naive approach maintains less accuracy than our algorithm with both MSE and STD measures except for the RR category. The reason is that for RR region, they require more types of characterizations to featurize their traffic flow.

5.2.3 Confidence Interval Evaluation

In this set of experiments, we use confidence intervals (CI) to indicate the reliability of our estimates. In particular, we evaluate the intensity of the featured clusters generated by the algorithms using CI. We consider the level of confidence interval is 90%, and use the mean of all distances between the instances and their cluster centroid as the observed mean value. Therefore, the lower mean value we get, the denser the cluster is. Figure 7 depicts the Euclidean distance between the instances and the cluster centroids (Y-axis) for eight spatial categories (X-axis). As illustrated, the naive algorithm has more sparse population of instances in each category.

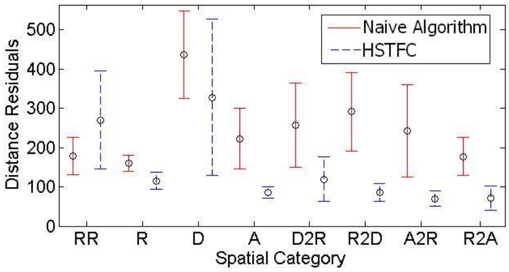


Figure 7: Confidence interval evaluation

6. CONCLUSION AND FUTURE WORK

In this paper, we introduced a framework for realistic and accurate modeling of traffic flows in spatiotemporal networks. We explained the design and implementation of our framework based on a real-word traffic sensor dataset. We intend to pursue this work in three directions. First, we plan to extend the set of spatial characteristics supported by our framework to a complete minimum set that allows for modeling all typical spatial networks. Second, we plan to incorporate temporal characteristics (e.g., congestion intervals) of the spatial networks to our framework. Third, we plan to design efficient query processing algorithms (e.g., nearest neighbor, range) on spatiotemporal networks since commonly assumed techniques on spatial networks would not hold for spatiotemporal networks.

7. ACKNOWLEDGMENTS

This research has been funded in part by NSF grants IIS-0238560 (PECASE), IIS-0534761, and CNS-0831505 (CyberTrust),the NSF Center for Embedded Networked Sensing (CCR-0120778) and in part from the METRANS Transportation Center, under grants from USDOT and Caltrans. Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

8. REFERENCES

[1] P. Athol. Interdependence of certain operational characteristics within a moving traffic stream. In *TRB*, 1967.

[2] T. Brinkhoff. A framework for generating network-based moving objects. In *Geoinformatica*, 2002.

[3] U. Demiryurek, F. B. Kashani, and C. Shahabi. Efficient continuous nearest neighbor query in spatial networks using euclidean restriction. In *SSTD*, 2009.

[4] B. Ding, J. X. Yu, and L. Qin. Finding time-dependent shortest paths over large graphs. In *EDBT*, 2008.

[5] B. George, S. Kim, and S. Shekhar. Spatio-temporal network databases and routing algorithms: A summary of results. In *SSTD*, 2007.

[6] S. P. Hoogendoorn and P. Bovy. State-of-the-art of vehicular traffic flow modelling. In *Journal of Systems and Control Engineering*, 2001.

[7] Y. Kamarianakis and P. Prastacos. Space-time modeling of traffic flow. 2007.

[8] E. Kanoulas, Y. Du, T. Xia, and D. Zhang. Finding fastest paths on a road network with speed patterns. In *ICDE*, 2006.

[9] H. v. Lint, S. P. Hoogendoorn, and H. J. v. Zuylen. State space neural networks for freeway travel time prediction. In *ICANN*, London, UK, 2002.

[10] K. Mouratidis, M. L. Yiu, D. Papadias, and N. Mamoulis. Continuous nearest neighbor monitoring in road networks. In *VLDB*, 2006.

[11] Navteq. http://www.navteq.com. Last visited June 17, 2009.

[12] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query processing in spatial network databases. In *VLDB*, 2003.

[13] PeMS. https://pems.eecs.berkeley.edu/. Last visited May 15, 2009.

[14] M. Pursula. Simulation of traffic systems-an overview. In *Journal of GIS and Decision Analysis*, 1999.

[15] RIITS. http://www.riits.net/. Last visited December 25, 2008.

[16] H. Samet, J. Sankaranarayanan, and H. Alborzi. Scalable network distance browsing in spatial databases. In *SIGMOD*, 2008.

[17] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. K-means clustering with background knowledge. In *ICML*, 2001.