Clustering Speed in Multi-lane Traffic Networks

Bing Zhang[*]
bing@u.northwestern.edu

Goce Trajcevski[†]
goce@eecs.northwestern.edu

Feiying Liu
feiying@u.northwestern.edu

Northwestern University
2145 Sheridan Road
Evanston, IL, 60201

ABSTRACT

We address the problem of efficient spatio-temporal clustering of speed data in road segments with multiple lanes. We postulate that the navigation/route plans typically reported by different providers as a single-value need not be accurate in multi-lane networks. Our methodology generates lane-aware distribution of speed from GPS data and agglomerates the basic space and time units into larger clusters. Thus, we achieve a compact description of speed variations which can be subsequently used for more accurate trips planning. We provide experiments that demonstrate the benefits of our proposed approaches.

Keywords

Speed Clustering, Trajectories, Multi-lane Roads

1. INTRODUCTION

Routing and navigation have been popular applications relying on traffic data since 1980s [6]. Techniques for efficiently determining driving directions in terms of shortest travelled distance or shortest travel-time, or even eco-routes (i.e., in terms of optimizing the fuel consumption and/or CO emission) have been investigated by many researchers, both in databases [12] and transportation communities [9], and implemented by various map services providers (e.g., Google, HERE, etc.). Algorithmic calculations rely on certain estimated values of the traffic flow – equivalently, average speed along the segments of the underlying road-networks – a value that changes dynamically [12, 13] within a certain period (e.g., a day) and may depend on several factors, such as: time of day, capacity (lanes), road surface, etc. Traditionally, traffic speed estimation is based on an averaged speed among all observed vehicles as a "characteristic value", for which there are two main approaches: (1) Speed samples from each vehicle at different timestamps (averaged within road segment) [8]; (2) Total travel distance divided by the total travel time along a particular/targeted road segment [12]. Existing route-planning methodologies and tools share one basic assumption: on any road segment, at a certain time-period, vehicles have only one kind of motion. However, the existence of multiple lanes may yield significantly different speeds on the same road-segment and at same time-intervals. Thus, using the average speed as a descriptor may not be good enough for many routing applications. As an illustration, consider an actual picture from Chicago's Lake Shore Drive, near Belmont Avenue exit, shown in Figure 1. The cars back up at the rightmost lane by Belmont Avenue exit – used when accessing Wrigley Field baseball stadium, while the rest of the 3 lanes on northbound Lake Shore have very low densities. This situation occurs quite frequently during non-rush hours throughout the baseball season. Thus, averaging the observations from particular (groups of) vehicles, could produce an inaccurate picture about the traffic distribution – and, yet, most of the popular traffic speed estimation methods are based on averaging the samples from vehicles over a period of time or area — e.g., *Time Mean Speed* and *Space Mean Speed* [4].

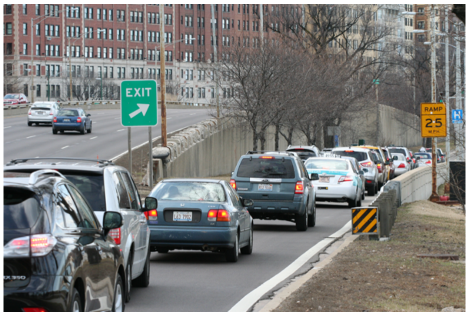

Figure 1: Belmont Ave. exit on Lake Shore Drive

At the heart of the motivation for this work is the observation that – to the best of our knowledge – the state of the art approaches have not incorporated multi-lane information when designing traffic speed profiles to be used for routes planning. We note that the research on map-matching GPS points from moving objects [2] has not fully exploited the settings of multiple lanes.

As mentioned, route planning is a foundational task for many Location-Based Services but traditional methods [8, 12] can only vary a single (average) speed value throughout: (a) different time intervals; and (b) along different (portions

[*] Research Supported by the NSF III 1213038 grant

[†] Research Supported by the NSF grants CNS 0910952 and III 1213038, and ONR grant N00014-14-10215

CIKM'16, October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983905

of) road segments. In the context of the example in Figure 1, this would mean that a request for a route-plan between Navy Pier and Lawrence Avenue would use the average speed around Belmont Ave. exit, yielding incorrect time-estimate for the targeted trip planning. To tackle such issues, our main contributions are:

- We achieve a compact description of speed variations in multi-lane road networks that can be used for more accurate trips planning.
- We propose an agglomerative speed clustering algorithm to represent the distribution in a more compact manner.
- We present experimental observations based on real data from highways around Rome to demonstrate the benefits of our proposed method.

2. PRELIMINARIES

We now present a brief overview of the related background and introduce the terminology. Traditionally, in Moving Objects Databases (MOD) [5], the motion of an object with a distinct ID (oID) is represented as a *trajectory* $Tr_{oID} = [p_1, p_2 ... p_n]$, where each p_i is a triplet $p_i = (x_i, y_i, t_i)$; t_i being the time that the object was at location (x_i, y_i).

In this work, a road-network is a collection of multi-lane road segments. A *road segment* r is a octuple $r = (r_{ID}, r_{Dir}, r_s, r_e, r_{type}, r_{length}, r_{speed}, r_{lane})$, where: r_{ID} is its unique identifier; r_{Dir} is a binary value indicating whether r is one-way or two-way segment; r_s and r_e are k-tuples ($k =$ number of lanes) representing the starting and ending points of each lane (centroids); r_{type} indicates the type of the road to which the segment belongs (e.g., urban, rural, etc...); r_{length} is its length; r_{speed} is the maximum speed; and r_{lane} is an integer specifying the number of lanes in each direction. A *road network* is an (augmented) graph $G_{RN} = (V_{RN}, E_{RN})$ where V_{RN} is the set of nodes representing the terminal points of road segments, and E_{RN} is the collection of road segments.

We note that in traditional traffic-stream studies there are different measures characterizing the motion along road segments [4], often coupled with the available technology. For example, inductive sensors are good for estimating the flow, however, they cannot characterize the speed. On-board GPS devices are good at obtaining an average speed of individual moving objects, however, they are error-prone in terms of location, and cannot capture fluctuations in-between samples. In this work, we assume that motion-relevant data is obtained from (a sequence of) GPS points via a suitable map-matching algorithm.

3. SPATIAL-TEMPORAL PARTITIONING AND MERGING

Partition-and-merge framework for clustering trajectories has been proposed in [7] – however, the work did not consider the time-dimension (i.e., the speed) and was dealing with free 2D motion, not constrained to road networks. Models have been proposed based on trajectories' geographical information (including moving objects' heading and trajectory density) and semantic information. Compared with conventional trajectory clustering framework, in our problem of mining traffic speed clusters, GPS sample points are constrained by road-network. However, map-matching with uncertain location poses other challenges.

There are three steps for data partitioning: a) map-matching GPS points to corresponding road segments. b) Identifying lane information within augmented road-network. c) Spatial-temporal partitioning to divide road segments and associated GPS points into basic units for further merging.

3.1 Map-matching and Lane Labeling

Map-matching algorithms use information generated from positioning technologies and supplement it with data from a high resolution road-network map to provide an enhanced positioning output. They identify the correct road on which vehicles travel and determine vehicles' location on that segment – and can be categorized into four groups: geometric, topological, probabilistic, and other advanced techniques [10]. In this work, we extend the most commonly used geometric map-matching algorithm called "point-based matching" (or point-to-curve matching).

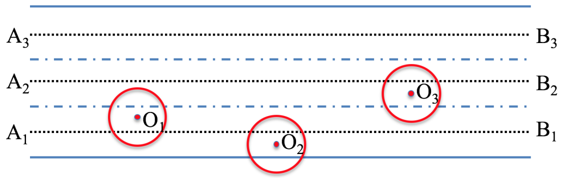

Figure 2: Lane labeling

An augmented road segment and three GPS sample points are shown in figure 2. Traditional road segment without width is merely the central line $\overline{A_2B_2}$. We augment it with lane width – yielding $\overline{A_1B_1}$ and $\overline{A_3B_3}$ as central line of inner lane and outer lane accordingly.

The uncertainty area of each GPS sample point is modeled by a horizontal disk with radius r centered at (x, y, t), where (x, y) is the expected location at time $t \in (t_i, t_{i+1})$ [11]. Let $D_p(x, y, t, r)$ denote the disk centered at point $P(x, y, t)$ with radius r, and A_i denote the area of lane i, approximated by a rectangle. Assuming an uniform distribution of the location inside D_p, the probability of one GPS point located within the area of certain lane can be estimated by: $P_{lane} = \frac{D_p(x,y,t,r) \cap A}{D_p(x,y,t,r)}$. Then, GPS points are matched to the lane ID which has the highest probability.

3.2 Partitioning and Merging

The philosophy of clustering traffic speed data is to group those GPS points that are spatially and temporally close to each other *and* with similar speed. Thus, based on a suitable minimum unit, larger clusters can be formed through merging and reorganizing. We define the Unit Cell (UC) as:
Unit Cell: A Unit Cell $UC_{kl} = (\Delta^S_{kl}, \Delta^T_{kl}, V_{kl}, D_{kl})$ in the l^{th} lane of a given road segment is the minimal partition in spatial and temporal dimension, characterized by a spatial range $\Delta^S_{kl} = d^+_{kl} - d^-_{kl}$, temporal interval $\Delta^T_{kl} = t^+_{kl} - t^-_{kl}$, and a set of trajectories $D_{kl} = [Tr_1, Tr_2, ... Tr_n]$ that belong to it. The set D_{kl} determines the speed-value V_{kl} associated with UC_{kl} – i.e., it is a set of trajectories such that they all have values of their speed that is within $[v^{min}_{k,l}, v^{max}_{k,l}]$ throughout the spatial and temporal boundaries of UC_{kl}. More importantly, $v^{max}_{k,l} - v^{min}_{k,l} \leq \delta'_v$ – tolerance threshold – and $V_{kl} = (v^{max}_{k,l} + v^{min}_{k,l})/2$. We note that the spatial range uses only "1D interval" – i.e., $d^+_{kl} - d^-_{kl}$ because the

"conventional" 2-D space is restructured as 1-D along the driving direction. However, this is only representing the distance(s) from starting point of the road segment (i.e., the corresponding lane) until the beginning of the k-th unit cell and the width is pre-determined by the width of the lanes for a given road-type. Figure 3 shows a scenario of six such unit cells distributed in two lanes, splitting each lane in 3 components, and with one temporal component.

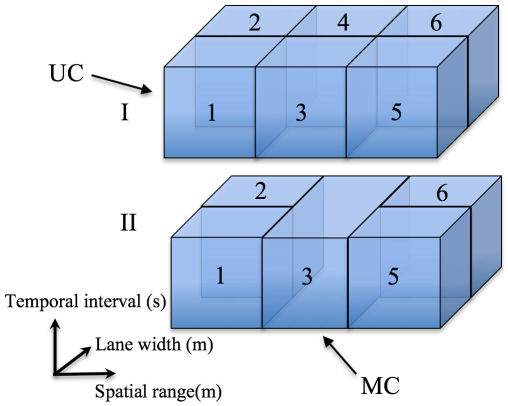

Figure 3: Merging mechanism

Merged Cell: Intuitively, a Merged Cell (MC) is a union of multiple neighboring unit cells $MC_j = UC_1 \cup UC_2 \cup ... \cup UC_n$. Its spatial range is defined as $R_{MC} = \cup_i \Delta_i^S$ and its temporal interval $T_{MC} = \cup_i \Delta_i^T$.

However, the more important criteria that we require for merging two neighboring unit cells, inspired by [3, 7], are:
(1) $V_{(k+1)l} - V_{kl} \leq \delta_v''$ – i.e., the speed-values in the cells are close enough to each other, and
(2) $||D_{(k+1)l}| - |D_{kl}|| \leq \tau$ – i.e., the cells need to have close enough number of trajectories in their support-set.

An illustration of a merging cell MC formed by two neighboring UCs is shown in figure 3, in the (*space, time, lane#*) dimensionality. We constrain the union to consist of neighboring UC's in order to keep the spatial and temporal continuity. The first observation that, depending on the merging order chosen, a given collection of *UCs* need not yield a unique (collection of) *MCs*. A slight generalization of the scenario illustrated in Figure 3 can easily demonstrate that a particular cell can participate in as many as six different mergings, provided that there are ≥ 3 lanes. One can envision cells as being nodes in a graph and edges existing between neighboring cells. Upon merging, two nodes coalesce into one – and, to select a criterion for merging, a priority needs to be assigned among the adjacent vertices. Figure 3 illustrates merging of two neighboring edges from two lanes, sharing the same time-interval and distance from the start of the road-segment. However, in the current implementation, this is the 3rd criterion:
(3.1) First we check whether cells can be merged along the spatial dimension within the same lane.
(3.2) If not, we check next whether two cells with the same spatial extent can be merged along the temporal dimension.
(3.3) Lastly, we check whether two cells can be merged along neighboring lanes.

For every iteration of merging process there are three basic steps that are followed when processing each node.

I If there are no neighbors along the merging direction, the current cell is skipped and marked as visited.

II We check the cross-section coordinates between current cell and its neighbor in the order of preferences of merging directions. If cross-section coordinates are aligned and they satisfy the merging criteria, a merged cell is formed to replace them. One example is shown in figure 3–I. When we merge UC_3 with UC_4 along lane width dimension, their cross-section are aligned and a new MC_3 is formed. If we try to merge MC_3 with UC_5 along spatial range dimension in figure 3–II, their cross-section cannot match, and the cell will be skipped.

III Newly merged cell inherits all the neighboring relationship from merged cells. At the same time, all neighbors of two merged cells update their neighbor lists, by replacing original cells with the new one.

Speed Cluster: When the merging process for a particular (unit or merged) cell can no longer continue, we call that cell a *Speed Clusters* (SC)

Algorithm 1 formalizes the above description.

Algorithm 1 Cluster Mining (CellSets, DirectionSets)

```
ClusterID = NextID(NULL);
for Merging Direction MD IN DirectionList do
    while Cell in CellSets is unvisited do
        Neighbor = Cell.Neighbors(i) along MD;
        if Neighbor.size() == 0 then
            Cell.visited = True;
            Continue;
        else if Align(Cell, Neighbor) == True AND CanMerge(Cell, Neighbor) == True then
            NewCell = merge(Cell, Neighbor);
            NewCell.Neighbors = mergeNeighbor(Cell, Neighbor);
            updateNeighbor(Cell, Neighbor, NewCell);
            delete Cell, Neighbor;
        else
            Cell.visited = True;
        end if
    end while
    CellSets.visited = False;
end for
Return CellSets;
```

Assume that there are a total of n GPS points in the database, and (on the average) a road segment is composed of $O(K)$ spatial intervals in each lane and $O(M)$ intervals in the temporal dimension, defining the unit cells. Under a uniform distribution, each unit cell will consist of $O(n/(KM))$ GPS points from various trajectories. Calculating the merged cells and clusters along a road segment can have an upper bound of $O((KM)^2)$ provided each one of the $K \cdot M$ cells is taken as a starting point to obtain the best possible clustering in terms of the minimal final number of clusters.

4. EXPERIMENTS AND CONCLUDING REMARKS

Dataset: The Grande Raccordo Anulare (GRA) is a toll-free, ring-shaped orbital motorway that encircles Rome. It is one of the most important roads with heavy traffic for the most of the day. The dataset contains GPS trace of 320 taxi cabs in Rome, collected over 30 days in the Rome area, from February 1 to March 2 of 2014 [1]. The cardinality of the dataset is 8,368,858.

Baseline: The baseline approach we compared with is a traditional traffic speed estimation method (cf. Section 1), by calculating the average speed among all speed samples. In order to incorporate the variation along temporal dimension, we divide speed samples into 24 subsets, which represent 24 hours in a day. Average speed within each hour is calculated accordingly.

Evaluation: The data was divided into four folds according to sample time, each containing GPS points within one week. We used three weeks data to train our model, and we randomly pick trajectories from the remaining set to estimate the travel time, by using both baseline method and traffic speed clustering model.

The experiment was repeated 15 times and the average of all the runs is shown in Figure 4. The x-axis represents the aggregated travel distance, and y-axis stands for average estimated travel. On average, the travel time by using speed clustering method is reduced by 20%, compared with baseline approach. For the entire road networks, there are 8,328,960 UCs before merging, and 414,982 MCs after merging, which is a 95% reduction.

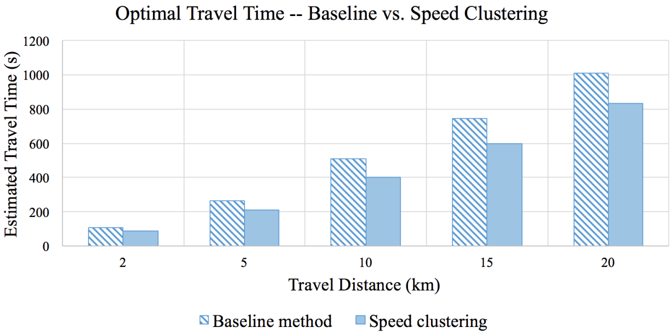

Figure 4: Travel time estimation using baseline approach and agglomerative method

We proposed a speed clustering approach that incorporate multi-lane information with partitioning and merging. We postulate that incorporating such information will yield a more precise calculation for route planning. We use Rome taxi data to demonstrate that, compared with baseline approach, our method yields faster trajectories for longer routes.

There are multiple extensions to our work. Firstly, we need to investigate the trade-offs in terms of complexity overheads (both space and time). Secondly, we note that partitioning and merging requires a choice of (fixed) merging rule. Even if this choice can be determined through experiment, additional computation is made. We plan to consider other kinds of learning approaches in conjunction with different clustering techniques. We also plan for a more thorough treatment of the (impact of the) GPS uncertainty/error [14] to the speed cluster mining algorithm. During the lane labeling process, large GPS errors will incur potential mislabeling, which later lead to noise points in the merging phase. One possible direction is to soft allocate the weight of GPS points to different lanes based on certain probabilities.

5. REFERENCES

[1] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from http://crawdad.org/roma/taxi/20140717, July 2014.

[2] R. Dalumpines and D. M. Scott. Gis-based map-matching: Development and demonstration of a postprocessing map-matching algorithm for transportation research. In *Advancing geoinformation science for a changing world*, pages 101–120. Springer, 2011.

[3] C. H. Q. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proceedings of the 2001 IEEE ICDM*, pages 107–114, 2001.

[4] N. H. Gartner, C. J. Messer, and A. K. Rathi. *Monograph on traffic flow theory.* Federal Highway Administration, 1997.

[5] R. H. Güting and M. Schneider. *Moving Objects Databases.* Morgan Kaufmann, 2005.

[6] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(3):345–358, 1992.

[7] J.-G. Lee, J. Han, X. Li, and H. Gonzalez. Traclass: trajectory classification using hierarchical region-based and trajectory-based clustering. *Proceedings of the VLDB Endowment*, 1(1):1081–1094, 2008.

[8] A. Pascale, F. Deflorio, M. Nicoli, B. Dalla Chiara, and M. Pedroli. Motorway speed pattern identification from floating vehicle data for freight applications. *Transportation Research Part C: Emerging Technologies*, 51:104–119, 2015.

[9] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

[10] M. a. Quddus, W. Y. Ochieng, and R. B. Noland. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies*, 15:312–328, 2007.

[11] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing uncertainty in moving objects databases. *ACM Trans. Database Syst.*, 29(3):463–507, Sept. 2004.

[12] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD*, pages 25–34. ACM, 2014.

[13] J. Yuan, Y. Zheng, X. Xie, and G. Sun. T-drive: enhancing driving directions with taxi drivers' intelligence. *Knowledge and Data Engineering, IEEE Transactions on*, 25(1):220–232, 2013.

[14] B. Zhang, G. Trajcevski, and L. Liu. Towards fusing uncertain location data from heterogeneous sources. *GeoInformatica*, 20(2):179–212, 2016.