On Splitting Raw Trajectories

Areeg Mostafa
mosta041@umn.edu
University of Minnesota, USA

Mohamed F. Mokbel
mokbel@umn.edu
University of Minnesota, USA

Ana Elena Uribe
uribe055@umn.edu
University of Minnesota, USA

Abstract

With the surge of data-driven solutions for trajectory analysis operations, the need for accurate trajectory trip data has spiked. However, the available datasets are raw trajectories spanning from hours to years, not representing actual trips for downstream applications. Therefore, pre-processing steps, such as basic rules to extract trips, are needed to use the datasets. However, this paper demonstrates that the current pre-processing steps are not enough and result in low accuracy, negatively affecting the downstream applications. This paper presents an overview of an accurate and scalable algorithm for splitting raw trajectories for trip extraction. We go beyond the basic rules to introduce a realistic definition of a trip and offer two scalable heuristics over the exhaustive brute force approach of the algorithm with similar accuracy. Experimental results show that the proposed algorithm is: (a) far more accurate than the basic rules, (b) scalable when employing either of the heuristics.

CCS Concepts

• **Information systems** → *Location based services.*

Keywords

Trajectory, Trips, Segmentation, Spatial-temporal

ACM Reference Format:
Areeg Mostafa, Mohamed F. Mokbel, and Ana Elena Uribe. 2024. On Splitting Raw Trajectories. In *The 32nd ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '24), October 29-November 1, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3678717.3691258

1 Introduction

The wide spread of internet of things, location detection technology, and location-tracking devices have enabled the means to collect real trajectory movement data where many are publicly available (e.g., [4, 7, 13, 24, 27, 29, 38, 44]). There is even an industry that sells such datasets (e.g., [28, 35, 36]). The availability of such data has empowered a myriad of trajectory analysis needs (e.g., see [40, 48] for surveys) that are deemed essential for various important applications and domains, e.g., transportation [12, 26], location-based services [1, 21, 23, 49], urban planning [32, 42], the health domain [3, 33], and evaluating trajectory analysis techniques [16–19, 45].

Despite their wide availability and usefulness for various applications, a large ratio of available trajectory data is not suitable to such applications. In particular, many trajectory datasets (e.g., [29, 34]) are released in their raw form as a sequence of locations spanning anywhere between hours and years. For example, the San Francisco Municipal Transportation trajectory dataset [29] is composed of 1,237 vehicles trajectories, each spanning a whole year. Such trajectories are not helpful to a wide spectrum of trajectory analysis applications that need to have access to *meaningful* trips. Examples of such applications would include anything that needs access to an Origin-Destination (OD) matrix, estimated time of arrival (ETA), routing, and trajectory similarity. However, the raw form of trajectories is composed of a set of back-to-back trips and unnecessary idle time. Hence, there is an immense need to extract meaningful trips from any raw trajectory. In practice, a pre-processing step needs to be applied before using the dataset. Commonly, this includes applying a set of *basic* rules to split each raw trajectory into individual trips. Such rules are typically about spatial, temporal, and speed constraints. For example, two consecutive GPS points have a significant spatial or temporal gap (more than a certain threshold), then they belong to two different trips. Such gaps represent the case of when the location-detection device was either not sending information or sending erroneous locations. Another rule is if a set of consecutive GPS points are in the same location, the trip will be ended at the first point in the set, duplicates will be removed, and the end point will be the start of a new trip. Such rule represents a significant idle time, e.g., parking.

This paper makes the case that these basic rules are not enough and would degrade the accuracy of downstream operations, as we define a *meaningful* trip to be a single destination trip without intermediate stops. For example, in the Estimated Time of Arrival (ETA) application, we can use the duration of a single destination as is and it will be accurate. However, if we use the duration of a multiple destination trip, because it was not detected by the basic rules, then it will be inaccurate and will give the illusion of traffic congestion between the start and end of the trip. This scenario can be wrongly interpreted by any algorithm as a traffic congestion. Unfortunately, multiple destination trips (e.g., back-to-back taxi trips) are undetectable using the basic rules as they do not have spatial or temporal gaps, or idle points, which negatively affects any historical trajectory-dependent operation or application (e.g., data-driven routing [14, 47], travel time estimation [41, 46], and edge weight inference [31, 43]). Such operations would only work accurately if fed with single-destination trips.

The work of this paper is supported by the National Science Foundation (NSF), USA, under grant IIS-2203553.

SIGSPATIAL '24, October 29-November 1, 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1107-7/24/10
https://doi.org/10.1145/3678717.3691258

The rest of the paper is organized as follows: Section 2 highlights related work. Section 3 introduces our objective function for trajectory splitting. A brute force approach to achieve the objective function is introduced in Section 4. Two scalable heuristics are presented in Section 5. Section 6 presents our experimental evaluation, while Section 7 concludes the paper.

2 Related Work

The process of splitting a long raw trajectory into a set of sub trajectories have been done for different purposes and with different titles, including trajectory segmentation [9] and trip extraction [37]. The body of work in this domain can be broadly categorized into the three below categories:

Rule-based Trajectory Segmentation. Working with raw trajectories, researchers have used some basic rules to segment each trajectory into a set of more realistic trajectories based on either finding outlier points [9], discovering idle states [10], or finding speeds exceeding a certain threshold [5]. This has then become conventional wisdom and a preprocessing step for any trajectory operation using raw trajectories. Follow up work has added more strict rules that are applicable to flight and animal trajectories [11, 37]. However, such additional rules are not suitable for vehicle trajectories, which is the focus of this paper. Applying such additional rules to vehicles would result in *over cutting*, i.e., splitting the raw trajectory into too many unnecessary splits. For example, while a sharp u-turn would be allowed in a vehicle trajectory, it is not for flight trajectories. Hence, applying such rule would result in an unnecessary split (i.e., over cut) of the raw vehicle trajectories, which will degrade the accuracy of downstream operations.

Modality-based Trajectory Segmentation. This category of work aims to segment personal trajectories based on their modality, e.g., walking, biking, bus, and vehicle. This is the most common purpose of segmentation, as each type of trajectories can be processed individually for a different purpose, e.g., walking trajectories can help in understanding customers' behavior while vehicle trajectories can help in traffic studies. There have been numerous approaches for such segmentation and many of them have employed various machine learning approaches to learn form historical data [6, 20, 39]. Other approaches use rule-based measures to separate each modality [30, 40]. None of these techniques are applicable to the case of trip extraction from vehicle trajectories. In fact, employing modality-based trajectory segmentation to separate vehicle trajectories may result in long *raw* vehicle trajectories that would still need to be split into individual single-destination trips.

Behavior-based Trajectory Segmentation. This is used for extracting human trips with different purposes or extracting the acceleration patterns for the same trip. A majority of the solutions utilize a rule-based approach to detect behavior changes and segment the trajectory accordingly. However, such segmentation is specific to the type of the trajectory being segmented in order to set the thresholds correctly [2]. For example, rules for human trajectories are based on human behavior [15], and rules for vehicle trajectories are based on driving behavior [22]. More recent generalized approaches use machine learning to learn and detect behavior changes with no underlying knowledge of the trajectory type [8]. Behavior change segmentation, even for trajectory vehicles, are not applicable to our case of trip extraction, as they tend to over cut, e.g., a single-destination trip with different driving behaviors in highway and residential area segments may be unnecessarily split.

3 Splitting Objective Function

This section sets our proposed objective function that any trajectory splitting algorithm should strive to achieve to produce realistic trajectory trips.

The Objective Function. *Given a raw trajectory T of N points, P_0 to P_{N-1}, split T into a set of trips S, where each trip in S starts at some trajectory point P_i and ends at P_j, and satisfies two properties: (1)* Property 1*: The actual travelled distance from P_i to P_j in trajectory T is within a deviation parameter $\alpha \geq 1$ from their shortest path road network distance, i.e., Distance(P_i,P_j) $\leq \alpha \times$ ShortestRoad(P_i,P_j), (2)* Property 2*. The end point P_j should satisfy* Property 1 *while point P_{j+1} should not. In effect, S should be as long as possible.*

The Deviation Parameter α. Drivers may not necessarily travel the shortest distance for various reasons. For example, drivers may have a personal route preference, or follow a routing service that suggested a longer, but faster, route. While this may be a common scenario, it is reasonable to assume that any route taken is still within a ratio from the shortest path. Hence, *Property 1* in our objective function introduces the deviation parameter α that controls what is considered as a reasonable, hence accurate, trip.

4 Brute Force Splitting

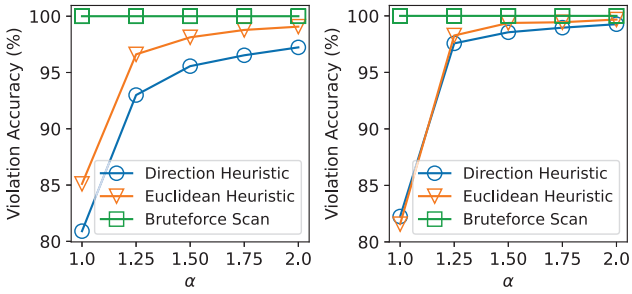
A brute force approach to achieve our proposed objective function is to simply go through each point in the raw trajectory and compute the *cumulative distance* of the current path and the *shortest path distance* from the starting point to the current point. To determine whether the current point is a splitting point, α is applied to the *shortest path distance* and if the *cumulative distance* is still larger, then the current point is a splitting point satisfying *Property 1*. Also, *Property 2* is satisfied, as all the raw trajectory points are scanned one by one, so, the brute force approach would be able to find the exact split point that will satisfy that property.

In terms of scalability, the brute force approach computes two shortest path distances for every trajectory point. This is computationally expensive and not scalable. For example, calling the shortest path function, twice per point, using OSRM API [25] takes 30 msec. Applying this for the whole San Francisco dataset [29] that has 500M trajectory points, reduced to 250M points by the *basic* rules that remove outliers and idle points, it would take 86 days to split all trajectories, which is not scalable for a preprocessing step.

5 Heuristic Splitting

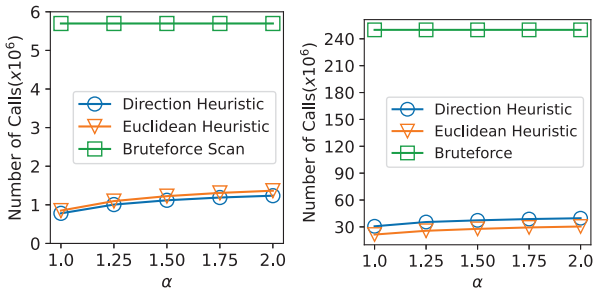
Due to the scalability challenges of the brute force approach, we are introducing two heuristics, namely, Direction and Euclidean heuristics, that aim to reduce the number of shortest path calls while maintaining the accuracy of the objective function.

Direction Heuristic. The main idea of the Direction-based Heuristic is to use the eight cardinal directions (north, northeast, east, southeast, south, southwest, west, and northwest) as a means of speculating if the given trajectory is moving back towards the starting point of the current trip. The rationale is that if the direction from the starting point is opposite to the current direction, then it



(a) T-Drive Dataset (b) San Francisco Dataset

Figure 1: Point Accuracy Ratio (%)



(a) T-Drive Dataset (b) San Francisco Dataset

Figure 2: Number of Shortest Path Calls

is a strong indication that a deviation is happening and the current point is a potential splitting point. In this case, the shortest path calls are only performed when the two directions (direction form starting point and current direction) are opposite. Calling a shortest path here does not mean that this would be a splitting point. It just means that this is a candidate to be a split point, which may end up to be a true split point or not. The optimization here comes from the fact that we call the shortest path function only for those candidate points, rather than for all points as it was the case for the brute force approach. Note that for this heuristic, the eight cardinal directions can be replaced with any other direction convention.

Euclidean Heuristic. The Euclidean-based Heuristic uses the Euclidean distance to speculate if the trajectory is moving back to its starting point or not. If the Euclidean distance from the starting point to the current point is smaller than the previous distance, then there is a high chance that a deviation is occurring and the current point is considered as a candidate split point. Same like the case of the direction heuristic, the shortest path function is called only for those candidate points rather than all points, which would be a significant performance gain over the brute force approach.

6 Experiments

This section evaluates the performance of the proposed splitting heuristics from two aspects: (1) The accuracy of the two heuristics in terms of the ratio of points satisfying *Property 1* of the objective function, and (2) The scalability of the two heuristics in terms of their ability to support much larger datasets that cannot be supported by the brute force approach. All experiments in this section run using two public datasets: (1) San Francisco dataset [29], which include 500M time stamped GPS points for 1,237 transit vehicles trajectories, each spanning the full year of 2021, sampled every minute in San Francisco CA, USA. (2) T-Drive [34], which include 15M time stamped GPS points for 10,357 taxi trajectories spanning one week of February 2008, sampled every 5 minutes in Beijing, China. All shortest path computations for all algorithms are API calls to the Open Source Routing Machine (OSRM) [25]. All experiments are performed on a Linux server with 8 CPU@3.5GHz, 64 GB Memory and 2TB HDD.

Heuristics Accuracy. This experiment studies the accuracy of both the Direction and Euclidean heuristics. As we consider that the brute force approach of the algorithm as the most accurate approach, we are measuring the accuracy of the two heuristics as a percentage from the brute force approach. If we look at the ratio of points satisfying *Property 1* of the objective function in Figure 1, we find that both heuristics achieve over 95% accuracy for α values greater than 1.5, indicating that they are effective in adhering to our trip definition. The Euclidean heuristic slightly outperforms the Direction heuristic, especially in the T-Drive dataset, although differences are minimal for the large San Francisco dataset due to high sampling rates. Overall, both heuristics hold high accuracy, with Euclidean generally offering better performance.

Heuristics Scalability. This experiment examines the scalability of the Direction and Euclidean heuristics by comparing their efficiency to the exhaustive brute force approach. If we consider the number of shortest path calls for each heuristic in Figure 2, we find that in the T-Drive dataset, the brute force method requires nearly 6 million shortest path calls, unaffected by α. Both heuristics need about one-sixth of the calls compared to brute force, making them six times more scalable. As α increases, the number of calls slightly rises due to longer trips and more split candidates. For the San Francisco dataset, the brute force method requires nearly 250 million calls while the heuristics only needed between 30 to 40 million calls, making them six to eight times more scalable. Though the Euclidean heuristic is more sensitive to α than the Direction heuristic, overall, both heuristics are highly scalable compared to the brute force approach for all values of α.

7 Conclusion

This paper makes the case of the need to go beyond just applying a set of basic rules to split raw trajectories to actual trip trajectories. The paper then introduces a new definition of what would be counted as a realistic trip, as well as introducing a clear objective function that needs to be achieved to enure realistic trajectory splitting to actual trips. A brute force algorithm that strictly adheres to the proposed objective function is introduced, yet, it would need exhaustive computations, making it not scalable to large datasets. Therefore, two heuristics are proposed, Direction-based and Euclidean-based heuristics that aim to mimic the behavior of the brute force approach, yet, in a much more scalable way, without sacrificing the accuracy in terms of adhering to the main objective function. Experimental results based on two real datasets show that both heuristics are able to achieve more than 90% accuracy with significantly much less computations.

References

[1] S. Abbar, R. Stanojevic, M. Musleh, M. M. ElShrif, and M. F. Mokbel. A Demonstration of QARTA: An ML-based System for Accurate Map Services. *Proceedings of the VLDB Endowment*, 14(12):2723–2726, 2021.

[2] S. P. A. Alewijnse, K. Buchin, M. Buchin, A. Kölzsch, H. Kruckenberg, and M. A. Westenberg. A Framework for Trajectory Segmentation by Stable Criteria. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 351–360, Dallas, Texas, Nov. 2014.

[3] R. Alseghayer. Racoon: Rapid Contact Tracing of Moving Objects Using Smart Indexes. In *Proceedings of the IEEE International Conference on Mobile Data Management, MDM*, pages 274–276, June 2021.

[4] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from https://ieee-dataport.org/open-access/crawdad-romataxi, July 2014.

[5] M. Buchin, A. Driemel, M. J. van Kreveld, and V. Sacristán. Segmenting Trajectories: A Framework and Algorithms Using Spatiotemporal Criteria. *Journal of Spatial Information Science*, 3:33–63, 2011.

[6] R. D. Das and S. Winter. Automated Urban Travel Interpretation: A Bottom-up Approach for Trajectory Segmentation. *Sensors*, 16, 2016.

[7] D. Dias and L. H. M. K. Costa. CRAWDAD dataset coppe-ufrj/riobuses (v. 2018-03-19). Downloaded from https://ieee-dataport.org/open-access/crawdad-coppe-ufrjriobuses, Mar. 2018.

[8] M. Etemad, Z. Etemad, A. Soares, V. Bogorny, S. Matwin, and L. Torgo. Wise Sliding Window Segmentation: A Classification-Aided Approach for Trajectory Segmentation. In *Advances in Artificial Intelligence*, pages 208–219, Cham, May 2020.

[9] M. Etemad, A. S. Júnior, A. Hoseyni, J. Rose, and S. Matwin. A Trajectory Segmentation Algorithm Based on Interpolation-based Change Detection Strategies. In *EDBT/ICDT Workshops*, page 58, Mar. 2019.

[10] S. Guo, X. Li, W.-K. Ching, R. Dan, W.-K. Li, and Z. Zhang. GPS trajectory data segmentation based on probabilistic logic. *International Journal of Approximate Reasoning*, 103:227–247, 2018.

[11] R. Hariharan and K. Toyama. Project Lachesis: Parsing and Modeling Location Histories. In *GIScience*, pages 106–124, Berlin, Heidelberg, 2004.

[12] B. Hossain, K. A. Adnan, M. F. Rabbi, and M. E. Ali. Modelling Road Traffic Congestion from Trajectories. In *Proceedings of the ACM International Conference on Data Science and Information Technology, DSIT*, pages 117–122, July 2020.

[13] X. Huang, Y. Yin, S. Lim, G. Wang, B. Hu, J. Varadarajan, S. Zheng, A. Bulusu, and R. Zimmermann. Grab-posisi: An extensive real-life GPS trajectory dataset in southeast asia. In *Proceedings of the ACM SIGSPATIAL International Workshop on Prediction of Human Mobility, PredictGIS 2019*, pages 1–10, Chicago, IL, USA, Nov. 2019.

[14] C. S. Jensen. Value Creation from Massive Data in Transportation ? The Case of Vehicle Routing. *IEEE Data Engineering Bulletin*, 42(3):4–8, 2019.

[15] M. Kafsi, M. Grossglauser, and P. Thiran. Traveling Salesman in Reverse: Conditional Markov Entropy for Trajectory Segmentation. In *Proceedings of the IEEE International Conference on Data Mining, ICDM*, pages 201–210, Nov. 2015.

[16] A. Karatzoglou, A. Jablonski, and M. Beigl. A Seq2Seq Learning Approach for Modeling Semantic Trajectories and Predicting the Next Location. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 528–531, Nov. 2018.

[17] M. Laass, M. Kiermeier, and M. Werner. Improving Persistence Based Trajectory Simplification. In *Proceedings of the IEEE International Conference on Mobile Data Management, MDM*, pages 157–162, June 2021.

[18] A. Liu, Y. Zhang, X. Zhang, G. Liu, Y. Zhang, Z. Li, L. Zhao, Q. Li, and X. Zhou. Representation Learning With Multi-Level Attention for Activity Trajectory Similarity Computation. *IEEE Transactions on Knowledge and Data Engineering, TKDE*, 34(5):2387–2400, 2022.

[19] C. Long, R. C. Wong, and H. V. Jagadish. Trajectory Simplification: On Minimizing the Direction-based Error. *Proceedings of the International Conference on Very Large Data Bases, PVLDB*, 8(1):49–60, 2014.

[20] C. Markos, J. J. Q. Yu, and Y. D. R. Xu. Capturing Uncertainty in Unsupervised GPS Trajectory Segmentation Using Bayesian Deep Learning. *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, 35:390–398, 2021.

[21] M. F. Mokbel, W. G. Aref, S. E. Hambrusch, and S. Prabhakar. Towards Scalable Location-aware Services: Requirements and Research Issues. In *Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM GIS*, page 110–117, Nov. 2003.

[22] S. Moosavi, R. Ramnath, and A. Nandi. Discovery of Driving Patterns by Trajectory Segmentation. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–4, Burlingame, CA, USA, Oct. 2016.

[23] M. Musleh, S. Abbar, R. Stanojevic, and M. F. Mokbel. QARTA: An ML-based System for Accurate Map Services. *Proceedings of the International Conference on Very Large Data Bases, PVLDB*, 14(11):2273–2282, 2021.

[24] Kaggle. New York City Taxi Trip Duration. https://www.kaggle.com/c/nyc-taxi-trip-duration/data.

[25] Open Source Routing Machine (OSRM). http://project-osrm.org/.

[26] S. A. Pedersen, B. Yang, and C. S. Jensen. Anytime Stochastic Routing with Hybrid Learning. *Proceedings of the International Conference on Very Large Data Bases, PVLDB*, 13(9):1555–1567, 2020.

[27] Taxi Service Trajectory. Prediction Challenge. ECML PKDD 2015. https://archive.ics.uci.edu/dataset/339/taxi+service+trajectory+prediction+challenge+ecml+pkdd+2015.

[28] SafeGraph. Your Partner in Places Data. https://www.safegraph.com/.

[29] San Francisco Municipal Transportation Agency (SFMTA) - Transit Vehicle Location History (Current Year). https://data.sfgov.org/Transportation/SFMTA-Transit-Vehicle-Location-History-Current-Yea/x344-v6h6/about_data.

[30] Y. Shen, H. Dong, L. Jia, Y. Qin, F. Su, M. Wu, K. Liu, P. Li, and Z. Tian. A Method of Traffic Travel Status Segmentation Based on Position Trajectories. In *Proceedings of the IEEE International Intelligent Transportation Systems Conference, ITSC*, pages 2877–2882, Gran Canaria, Spain, Sept. 2015.

[31] R. Stanojevic, S. Abbar, and M. Mokbel. W-edge: Weighing the Edges of the Road Network. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 424–427, Seattle, WA, USA, Nov. 2018.

[32] R. Stanojevic, S. Abbar, S. Thirumuruganathan, S. Chawla, F. Filali, and A. Aleimat. Robust Road Map Inference through Network Alignment of Trajectories. In *Proceedings of the SIAM International Conference on Data Mining, SDM*, pages 135–143, May 2018.

[33] C. Sydora, F. Nawaz, L. Bindra, and E. Stroulia. Building Occupancy Simulation and Analysis under Virus Scenarios. *ACM Transactions on Spatial Algorithms and Systems, TSAS*, 8(3):1–20, 2022.

[34] T-Drive trajectory data sample. https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/.

[35] Unacast. Build Better Products and Make Smarter Decisions with Real-world Location Data. https://www.unacast.com/.

[36] Veraset. Your Trusted Partner for Mobility Data. https://www.veraset.com/.

[37] F. Wang, J. Wang, J. Cao, C. Chen, and X. J. Ban. Extracting Trips from Multi-Sourced Data for Mobility Pattern Analysis: An App-based Data Example. *Journal of Transportation Research Part C: Emerging Technologies*, 105:183–202, 2019.

[38] G. Wang, X. Chen, F. Zhang, Y. Wang, and D. Zhang. Experience: Understanding Long-Term Evolving Patterns of Shared Electric Vehicle Networks. In *Proceeding of the International Conference on Mobile Computing and Networking, MobiCom*, pages 1–12, Los Cabos, Mexico, Oct. 2019.

[39] L. Wang, W. Ma, Y. Fan, and Z. Zuo. Trip Chain Extraction using Smartphone-Collected Trajectory Data. *Transportmetrica B: Transport Dynamics*, 7:255–274, 2019.

[40] S. Wang, Z. Bao, J. S. Culpepper, and G. Cong. A Survey on Trajectory Data Management, Analytics, and Learning. *ACM Computing Surveys*, 54(2):39:1–39:36, 2021.

[41] N. Wu, J. Wang, W. X. Zhao, and Y. Jin. Learning to Effectively Estimate the Travel Time for Fastest Route Recommendation. In *Proceedings of the International Conference on Information and Knowledge Management, CIKM*, pages 1923–1932, New York, NY, USA, Nov. 2019.

[42] H. Xue, F. D. Salim, Y. Ren, and N. Oliver. MobTCast: Leveraging Auxiliary Trajectory Forecasting for Human Mobility Prediction. In *Proceedings of the Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 30380–30391, Dec. 2021.

[43] B. Yang, M. Kaul, and C. S. Jensen. Using incomplete information for complete weight annotation of road networks. *IEEE Transactions on Knowledge and Data Engineering, TKDE*, 26(5):1267–1279, 2014.

[44] Y. Yang, F. Zhang, and D. Zhang. SharedEdge: GPS-Free Fine-Grained Travel Time Estimation in State-Level Highway Systems. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):48:1–48:26, 2018.

[45] J. J. Ying, W. Lee, T. Weng, and V. S. Tseng. Semantic Trajectory Mining for Location Prediction. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34–43, Nov. 2011.

[46] H. Yuan, G. Li, and Z. Bao. Route Travel Time Estimation on a Road Network Revisited: Heterogeneity, Proximity, Periodicity and Dynamicity. *Proceedings of the International Conference on Very Large Data Bases, PVLDB*, 16(3):393–405, 2023.

[47] J. Yuan, Y. Zheng, X. Xie, and G. Sun. T-Drive: Enhancing Driving Directions with Taxi Drivers' Intelligence. *IEEE Transactions on Knowledge and Data Engineering, TKDE*, 25(1):220–232, 2013.

[48] Y. Zheng. Trajectory Data Mining: An Overview. *ACM Transactions on Intelligent Systems and Technology, TIST*, 6(3):29:1–29:41, 2015.

[49] F. Zhou, H. Wu, G. Trajcevski, A. A. Khokhar, and K. Zhang. Semi-supervised Trajectory Understanding with POI Attention for End-to-End Trip Recommendation. *ACM Transactions on Spatial Algorithms and Systems, TSAS*, 6(2):13:1–13:25, 2020.

Received 07 June 2024; revised 26 July 2024; accepted 23 August 2024