Bulletin of the Technical Committee on

Data Engineering

September 2019 Vol. 42 No. 3 IEEE Computer Society

The TC on Data Engineering

Membership in the TC on Data Engineering is open to all current members of the IEEE Computer Society who are interested in database systems. The TCDE web page is http://tab.computer.org/tcde/index.html.

The Data Engineering Bulletin

The Bulletin of the Technical Committee on Data Engineering is published quarterly and is distributed to all TC members. Its scope includes the design, implementation, modelling, theory and application of database systems and their technology.

Letters, conference information, and news should be sent to the Editor-in-Chief. Papers for each issue are solicited by and should be sent to the Associate Editor responsible for the issue.

Opinions expressed in contributions are those of the authors and do not necessarily reflect the positions of the TC on Data Engineering, the IEEE Computer Society, or the authors' organizations.

The Data Engineering Bulletin web site is at
http://tab.computer.org/tcde/bull_about.html.

Letter from the Editor-in-Chief

Letter from Haixun

Haixun Wang
WeWork Corporation

Letter from the TCDE Awards Committee Chair

Johannes Gehrke
Microsoft, USA

Letter from the Special Issue Editor

Letter from Alexandra Meliou

Alexandra Meliou
University of Massachusetts, Amherst

Value Creation from Massive Data in Transportation—The Case of Vehicle Routing

Christian S. Jensen
Aalborg University, Denmark

1 Introduction

Vehicular transportation will undergo profound change over the next decades, due to developments such as increasing mobility demands and increasingly autonomous driving. At the same time, rapidly increasing, massive volumes of data that capture the movements of vehicles are becoming available. In this setting, the current vehicle routing paradigm falls short, and we need new data-intensive paradigms. In a data-rich setting, travel costs such as travel time are modeled as time-varying distributions: at a single point in time, the time needed to traverse a road segment is given by a distribution. How can we best build, maintain, and use such distributions?

The travel cost of a route is obtained by convolving distributions that model the costs of the segments that make up the route. This process is expensive and yields inaccurate results when dependencies exist among the distributions. To avoid these problems, we need a path-centric paradigm, where costs are associated with arbitrary paths in a road network graph, not just with edges. This paradigm thrives on data: more data is expected to improve accuracy, but also efficiency. Next, massive trajectory data makes it possible to compute different travel costs in different contexts, e.g., for different drivers, by using different subsets of trajectories depending on the context. It is then no longer appropriate to assume that costs are available when routing starts; rather, we need an on-the-fly paradigm, where costs can be computed during routing. Key challenges include how to achieve efficiency and accuracy with sparse data. Finally, the above paradigms assume that the benefit, or cost, of a path is quantified. As an alternative, we envision a cost-oblivious paradigm, where the objective is to return routes that match the preferences of local, or expert, drivers without formalizing costs.

2 Background

Vehicular transportation is an inherent aspect of society and our lives: many people rely on vehicular transportation on a daily basis, we spend substantial time on transportation, and we are often forced to arrange our lives around traffic. As a reflection of this, society spends very substantial resources on enabling safe, reliable, clean, and inexpensive transportation. Due to a combination of interrelated developments, transportation will undergo profound changes in the years to come.

First, a range of key enabling technologies have reached levels of sophistication that make (semi-)autonomous vehicles possible. For example, Tesla cars already come with an autopilot that is a pre-cursor to autonomous driving, and virtually all major vehicle manufacturers are working to make autonomous cars. The state of affairs is similar to the one that applied to personal computing when Apple and Microsoft were created and the one that applied to the Internet when Google was founded. Second, the sharing economy trend is also gaining traction in relation to vehicular transportation, thus enabling better exploitation of under-utilized vehicles. For example, Uber enables transportation in private vehicles by private drivers. Online ridesharing services such as Lyft enable the sharing of trips. A large number of similar services exist across the globe. Next, other developments such as urbanization and the needs to combat air pollution and greenhouse gas emissions will also impact transportation. Many large cities are facing air quality problems, and the transportation sector is the second largest contributor to GHG emissions, trailing only the energy sector.

These increasingly pressing developments promise a perfect storm for transportation: While it is not clear exactly how this will play out, it is clear that transportation faces profound change. For example, Uber and similar services may eventually do away with under-paid drivers. When a person goes to a movie theater and cannot find parking, the driver may instead let the car serve as a self-driving taxi, thus making money instead of paying money for parking while watching a movie.

We are also witnessing a digitalization trend that is unprecedented in the history of humanity: We are increasingly instrumenting societal and industrial processes with networked sensors. As a result, we are accumulating massive volumes of data that capture the states of processes and that may be used for enabling rational, data-driven processes and data-driven decision making. This also applies to transportation. Vehicles are increasingly online, via smartphones or built-in connectivity, and they are equipped with global navigation satellite system (GNSS) positioning capabilities, e.g., Galileo, GPS, and Glonass, via smartphones or in-vehicle navigation systems. As a result, rapidly increasing volumes of vehicle data are becoming available. This data includes vehicle trajectory data, i.e., sequences of GNSS records that record time and location. This new data source captures transportation at a level of detail never seen before.

With the diffusion of smartphones and in-vehicle navigation devices, routing is now available to a very large fraction of the population on Earth. Indeed, the availability of routing is now taken for granted, and routing is used widely. Further, the advances in autonomous and semi-autonomous vehicles make it a safe bet that more and more routing decisions will be taken by machines using some form of routing service, rather than by people. Thus, the importance of routing will increase over the coming years.

The foundation for traditional routing was built at a time where little data was available. We contend that given the above observations, new foundations are needed to enable routing capable of effectively exploiting available data to enable efficient and accurate, high-resolution routing services.

3 New Routing Paradigms

Traditional Routing The setting that underlies traditional routing services is one where a road network is modeled as a weighted graph and where the weight of an edge captures the cost of traversing the road segment modeled by the edge. In this setting, a graph with real-valued edge weights, capturing, e.g., travel distance, is given and some routing algorithm is applied to identify a route from a source to a destination with the minimum sum of edge weights. More advanced edge weights that capture travel time are also considered. While many different routing algorithms exist for such weighted road-network graphs, the prototypical algorithm is Dijkstra's algorithm [1]; hence, we call this Dijkstra's paradigm. This paradigm is well suited for settings were little travel data is available. Notably, by assigning weights to the atomic paths, i.e., individual graph edges, the paradigm makes the best possible use of available data. However, we contend that this simple edge-centric paradigm is obsolete and hinders progress in settings were travel costs are extracted from trajectories. Dijkstra's paradigm falls short when it comes to exploiting massive volumes of trajectory data for enabling more accurate and higher-resolution routing.

Given a (source, destination)-pair and a departure time, a typical routing service computes one or more paths from the source to the destination with the fastest travel time as of the departure time. "High resolution" implies that travel times in a road network are modeled (i) at a fine temporal granularity, as traffic changes continuously and affects travel time, and (ii) as distributions, as different drivers may have different travel times even when driving on the same path at the same time, and as traffic is inherently unpredictable. Further high resolution implies that routing takes into account the particular context, e.g., the driver, yielding personalized routing, or weather conditions [2, 3, 4].

We envision three new routing paradigms that are capable of exploiting massive trajectory data to enable more accurate and higher-resolution routing services.

Path-centric paradigm In this paradigm, costs are associated with arbitrary paths in a road network graph, rather than just with edges. This avoids unnecessary fragmentation of trajectories and automatically enables detailed capture of dependencies as well as turning and waiting times at intersections. This paradigm thrives on data: the more trajectory data, the better the accuracy and resolution of the routing. Further, more data also promises more efficient routing, which is less intuitive. With this paradigm, the cost, e.g., travel time, of an arbitrary path is estimated from available costs of paths that intersect the path. Fewer costs have to be assembled than in the edge-centric paradigm. For example, with costs being probability distributions and a path containing 100 edges, convolution must be applied 99 times to assemble 100 distributions into one in Dijkstra's paradigm. With sufficient trajectory data, a path may be covered by a few long paths with costs in the path-centric paradigm. Thus, computing the path's cost will require only a few convolutions. Thus, this paradigm holds the potential to enable more efficient routing the more trajectory data that is available. In the extreme, computing the cost of an arbitrary path can be achieved by means of a lookup, with no need for convolution. Next, when using Dijkstra's algorithm, intuitively, when a search has reached a graph vertex, the lowest-cost path to reach that vertex is known and fixed; thus, all other paths for reaching the vertex can be disregarded, or pruned. In the new paradigm, the cost of reaching a vertex can change when the search proceeds from the vertex because a different set of path costs that reach into the past may be used. It may happen that the cost of the path used for reaching the vertex increases and that a lower-cost path now exists.

In the path centric-paradigm, the underlying data structure is no longer just a graph, as path weights need to be maintained, and the correctness of Dijkstra's algorithm is no longer guaranteed. In initial work [5, 6], we have taken first steps to define and explore some aspects of the path-centric paradigm. These studies confirm that the paradigm holds substantial promise and is "the right" paradigm when massive trajectory data is available.

On-the-fly paradigm Next, massive trajectory data makes it possible to compute different travel costs in different contexts, e.g., for different drivers, by using different subsets of trajectories depending on the context. In this setting, it is no longer appropriate to assume that precomputed costs are available when routing starts, which is the standard assumption. There are simply too many costs to compute and store, most of which will never be used. Instead, we need an on-the-fly paradigm, where costs can be computed during routing. When, during routing, we need to determine the cost distribution of an edge or a path, we need to retrieve the relevant parts of the available trajectories that contain useful cost information given the particular context considered. These parts are then used to form an accurate cost distribution. The retrieval task takes a path, the time-of-arrival at the path, and contextual information such as a user identifier and weather information as arguments. Then the task is to retrieve sub-trajectories that contain information relevant to these arguments. As a routing query should preferably take less than 100 milliseconds, it is very difficult to achieve the necessary efficiency, and indexing techniques are needed that go beyond existing techniques [7, 8, 9]. Another challenge is to determine which trajectories to actually use when computing the most accurate weight distributions. We have conducted preliminary studies focused on achieving better indexing [10] and understanding the accuracy problem [11, 12]. The studies indicate that the challenges are substantial.

Cost-oblivious paradigm The above paradigms rely on the same underlying assumption as does Dijkstra's paradigm: We use trajectory data for computing costs, and then we apply a routing algorithm to find lowest-cost paths. In essence, these paradigms only use trajectories for extracting costs such as travel time and GHG emissions [13]. However, trajectories contain much more information that could potentially be utilized for achieving better routing: Trajectories tell which routes drivers follow and seemingly prefer. This paradigm is behavioral in the sense that it aims to exploit this route-choice behavior. An earlier study [14] indicates that historical trajectories are better at predicting the route a driver will take from a source to a destination than is the route returned by a cost-based routing service. This study thus confirms that the cost-oblivious paradigm holds potential for enabling better routing. And again, this is a paradigm that is shaped to thrive on data: If enough data is available to cover all (source, destination)-pairs with trajectories, routing could be achieved by means of a lookup, with no need for a travel-cost based routing algorithm. We have already proposed a simple route-recommendation solution and have compared it with existing solutions [15]. These solutions do not contend well with sparse data. In addition, we have proposed a first attempt at making better use of sparse data [16] for path recommendation within this paradigm.

Synergies It is important to observe that specific routing solutions can be composed of elements from Dijkstra's paradigm and all three new paradigms. For example, a predominantly on-the-fly solution may rely on pre-computed edge weights as a fall-back; and if insufficient data is available to a cost-oblivious solution, some limited form of routing may be applied. Beyond this, the fleshing out of the three paradigms relies on the same experimental infrastructure, encompassing computing capabilities, software pipelines, data, and methodologies.

4 Summary

In a world with more than 2.5 billion smartphone users and about 1 billion cars, and where routing decisions are increasingly being made by machines, the line of research outlined here has the potential for very large societal impact. It literally holds the potential to make a difference for on the order of a billion users. High-quality routing has significant benefits. It can make transportation more predictable, an important property of a transportation system that reduces the need to "leave early" and thus the time spent on transportation. In addition, it may increase the capacity of an existing infrastructure by making each trip more efficient, making room for more trips, and by incentivizing drivers to "spread out" their trips, e.g., by quantifying the time saved by traveling before or after rush hour. Routing also holds the potential to reduce the GHG emissions per trip [17, 18]. Finally, the above coverage of problems related to the use of massive trajectory data for value creation in transportation is by no means exhaustive.

Acknowledgments I would like to thank the many hard-working colleagues with whom I have worked and am working to make progress on the topics described here.

References

[1] E. W. Dijkstra. *A note on two problems in connexion with graphs*. Numer. Math., vol. 1, no. 1, pp. 269–271, 1959.

[2] J. Letchner, J. Krumm and E. Horvitz. *Trip Router with Individualized Preferences (TRIP): Incorporating Personalization into Route Planning*. In AAAI, 2006.

[3] B. Yang, C. Guo, Y. Ma and C. S. Jensen. *Toward personalized, context-aware routing*. VLDB J, vol. 24, no. 2, pp. 297–318, 2015.

[4] O. Andersen and K. Torp. *A Data Model for Determining Weather's Impact on Travel Time*. In DEXA, 2016.

[5] J. Dai, B. Yang, C. Guo, C. S. Jensen and J. Hu. *Path Cost Distribution Estimation Using Trajectory Data*. PVLDB, vol. 10, no. 3, pp. 85–96, 2016.

[6] B. Yang, J. Dai, C. Guo, C. S. Jensen and J. Hu. *PACE: a PAth-CEntric paradigm for stochastic path finding*. VLDB J, vol. 27, no. 2, pp. 153–178, 2018.

[7] B. B. Krogh, N. Pelekis, Y. Theodoridis and K. Torp. *Path-based queries on trajectory data*. In SIGSPATIAL GIS, 2014.

[8] B. B. Krogh, C. S. Jensen and K. Torp. *Efficient in-memory indexing of network-constrained trajectories*. In SIGSPATIAL GIS, 2016.

[9] S. Koide, Y. Tadokoro, C. Xiao and Y. Ishikawa. *CiNCT: Compression and retrieval of massive vehicular trajectories via relative movement labeling*. In ICDE, 2018.

[10] R. Waury, C. S. Jensen, S. Koide, Y. Ishikawa, and C. Xiao. *Indexing Trajectories for Travel-Time Histogram Retrieval*. In EDBT 2019.

[11] R. Waury, J. Hu, B. Yang and C. S. Jensen. *Assessing the Accuracy Benefits of On-the-Fly Trajectory Selection in Fine-Grained Travel-Time Estimation*. In MDM, 2017.

[12] R. Waury, C. S. Jensen and K. Torp. *Adaptive Travel-Time Estimation: A Case for Custom Predicate Selection*. In MDM, 2018.

[13] C. Guo, B. Yang, O. Andersen, C. S. Jensen and K. Torp. *EcoMark 2.0: empowering eco-routing with vehicular environmental models and actual vehicle fuel consumption data* Geoinformatica, vol. 19, no. 3, pp. 567–599, 2015.

[14] V. Ceikute and C. S. Jensen. *Routing Service Quality - Local Driver Behavior Versus Routing Services*. In MDM, 2013.

[15] V. Ceikute and C. S. Jensen. *Vehicle Routing with User-Generated Trajectory Data* In MDM, 2015.

[16] C. Guo, B. Yang, J. Hu and C. S. Jensen. *Learning to Route with Sparse Trajectory Sets*. In ICDE, 2018.

[17] O. Andersen, C. S. Jensen, K. Torp and B. Yang. *EcoTour: Reducing the Environmental Footprint of Vehicles Using Eco-routes*. In MDM, 2013.

[18] C. Guo, B. Yang, O. Andersen, C. S. Jensen and K. Torp. *EcoSky: Reducing vehicular environmental impact through eco-routing*. In ICDE, 2015.

Letter from the Impact Award Winner

I was very happy and humbled to receive this year's TCDE Impact Award, with the citation "for contributions to spatial, temporal, and spatio-temporal data management." I would like to thank those who nominated me as well as the awards'd committee. Conducting research is very much a social, or collaborative, activity, and I have worked with many excellent colleagues on the three topics mentioned in the citation, and they deserve most of the credit for the results that I have contributed to achieving. I will mention some of them as I cover aspects of my research journey. I started out working on temporal databases and then later transitioned to working on spatial and spatio-temporal databases. To achieve some degree of brevity, I will offer an account of only some of the activities related to temporal data management. I thus start at the very beginning of my academic life.

The Early Years—Ph.D. Studies I received my M.Sc. degree in computer science from Aalborg University in 1988. At that time, the M.Sc. study had a formal duration of five and a half years and included two B.Sc. degrees (in my case, in Mathematics and Computer Science). The last half year was devoted to the M.Sc. thesis, but the mindset at the time was that you were not serious if you spent less than a year. Thus, having received the M.Sc. degree after six years of study, I received a scholarship to go and study for a Ph.D. for two and a half years anywhere in the world. All I needed to do was to write a thesis—the course requirements were already satisfied.

In early September 1988, I then arrived at Dulles Airport. My M.Sc. supervisor, Lars Mathiassen, now a professor at Georgia State University, had recommended that I study under the direction of Leo Mark, then a young faculty member at the University of Maryland. I still remember driving with Leo from Dulles to his house in the late evening with all the windows open in his (by Danish standards) huge and very American Chevy. An exciting journey had started.

A November 25, 1988 plan gave the following working title for my thesis: "A By-Relation Implemented Object Oriented Data Model Supporting Efficient Storage and Retrieval of Versions of Complex Objects in Engineering Applications." I started out looking at the versioning aspect, and this led to studies of support for transaction time, which I viewed as an ideal foundation for fine-grained version support. The eventual title of the thesis was "Towards the Realization of Transaction Time Database Systems," and I had become interested in temporal databases.

The Pursuit of Industrial Impact Having completed the Ph.D. studies and defended the thesis back in Denmark in January 1991, I packed up my car in Greenbelt, MD and drove cross-country to Tucson, AZ, where I was to work with the most visible temporal database researcher, Rick Snodgrass, then a young faculty member at the University of Arizona. I had received a faculty position at Aalborg University that allowed me to spend my first semester with Rick. Our interests matched very well, and we got off to a very good start. This turned into three more sabbaticals, in 1992, 1994, and 1999, where I also got the opportunity to work with Rick's students, Curtis Dyreson, Nick Kline, and Mike Soo.

The 1990s were exciting times in temporal databases. The field had witnessed a proliferation of temporal data models and query languages, almost to the point of each researcher having their own model and language. It was felt that this blocked industrial impact, and initiatives were taken to achieve a consensus temporal data model and query language. This resulted in the TSQL2 query language, which was designed by an 18-person committee led by Rick.

Pursuing the goal of achieving industrial impact, Rick subsequently was the main force behind attempts to standardize TSQL2. This turned out to be a difficult process, in part due to politics and a variety of interests, but we also made technical progress. Specifically, we learned that the TSQL2 design approach did not scale well: Adding support for some temporal functionality to SQL worked fine, but adding comprehensive support following the TSQL2 approach was not pretty. While SQL is not a pretty language in the first place in terms of design, the TSQL2 approach yielded a result that was uglier than we would have liked. Something different was needed. As we were making these revelations, Michael Böhlen joined the University of Arizona as a postdoc. He had worked on an approach to language design that inspired the introduction of so-called statement modifiers into TSLQ2. The idea is that many temporal queries can be expressed intuitively and unambiguously as a single-state, non-temporal (and easy-to-formulate) SQL query that is then performed, as specified by a statement modifier, on all states of a temporal relation, after which the results are combined into a temporal relation. So a temporal query could then be formulated by a non-temporal query prefixed by some modifiers. A careful design based on this approach was introduced into standards proposals, and an "academic" version called ATSQL was also designed and documented in a TODS 2000 paper titled "Temporal Statement Modifiers."

In parallel with the above, I also worked on a range of other subjects in temporal databases, including database design, covering logical and conceptual temporal database design; data model and query language design aspects; support for the notion of "now" and for data aging; indexing; implementation of temporal algebra operators; query optimization; and architectures for implementing temporal query language support. I worked with five of my first six Ph.D. students on these topics: Kristian Torp, Heidi Gregersen, Dieter Pfoser, Janne Skyt, and Giedrius Slivinskas.

The Recent Years While spatial and spatio-temporal databases started to take over as my main activity around year 2000, I have continued to maintain an interest in temporal databases. Following his postdoc at Arizona, Mike joined the faculty at Aalborg University. He later moved to the Free University of Bozen-Bolzano and he is now back home in Switzerland, at the University of Zurich. I have been fortunate to be able to continue to work on temporal databases with Mike, Hans Gamper from Bolzano, and most recently Anton Dignös, as a Ph.D. student at Zurich and now as a faculty member at Bolzano. A key goal was to achieve an implementation of ATSQL. With other colleagues, we looked at many options, but it took until 2016, i.e., 16 years, before we had solid results. In particular, Anton's Ph.D. thesis and a TODS 2016 paper titled "Extending the Kernel of a Relational DBMS with Comprehensive Support for Sequenced Temporal Queries" show how to extend the kernel of PosgreSQL to enable efficient support for the functionality described in the TODS 2000 paper.

Impact and Lessons Looking back, one may ask what the impact of this work has been. Certainly, the literature suggests that the work has influenced other research in the field, but there has also been impact beyond academia. One highlight is that Teradata put temporal support into their system based on the statement modifier approach, which made them a pioneer in offering temporal support. This was done before ANSI/ISO standardization. Today, Teradata in addition supports the temporal tables and (limited) query language syntax in the standard. Another highlight is that the PostgreSQL implementation described in the TODS 2016 paper is available for anyone to use. A different line of impact is in the area of database design, where national statistics bureaus (e.g., Statistics Denmark) and archives (e.g., Danish National Archives) make use of temporal tables, including bi-temporal tables, when organizing their data. I have been contacted by, and have interacted with, several such entities. While the standards have adopted a language design approach that I think does not scale, and while there is a disconnect between SQL standardization and academia, I do believe that the standard is influenced by advances in temporal database research. For example, the standard supports bitemporal tables: We studied such tables in depth and even coined the term bitemporal.

Finally, I want to make a few points. First, research is often a social and collaborative effort. One should try to work with good colleagues (check!) and try to be a good colleague. Second, it can take decades to achieve societal impact, which is at odds with the increasing dependence on short externally funded projects in order to be able to perform research. Third, the disconnect between stardardization and academia is unfortunate from a societal perspective. Fourth, in research, one often does not quite know where one ends when starting.

Christian S. Jensen
Aalborg University, Denmark

Letter from the Service Award Winner

David Lomet
Microsoft Research, USA

Letter from the Rising Star Award Winner

Viktor Leis
Friedrich-Schiller-Universität Jena

Data Engineering

It's FREE to join!

TCDE

tab.computer.org/tcde/

The Technical Committee on Data Engineering (TCDE) of the IEEE Computer Society is concerned with the role of data in the design, development, management and utilization of information systems.

- Data Management Systems and Modern Hardware/Software Platforms
- Data Models, Data Integration, Semantics and Data Quality
- Spatial, Temporal, Graph, Scientific, Statistical and Multimedia Databases
- Data Mining, Data Warehousing, and OLAP
- Big Data, Streams and Clouds
- Information Management, Distribution, Mobility, and the WWW
- Data Security, Privacy and Trust
- Performance, Experiments, and Analysis of Data Systems

The TCDE sponsors the International Conference on Data Engineering (ICDE). It publishes a quarterly newsletter, the Data Engineering Bulletin. If you are a member of the IEEE Computer Society, you may join the TCDE and receive copies of the Data Engineering Bulletin without cost. There are approximately 1000 members of the TCDE.

Join TCDE via Online or Fax

ONLINE: Follow the instructions on this page:

www.computer.org/portal/web/tandc/joinatc

FAX: Complete your details and fax this form to **+61-7-3365 3248**

Name __________

IEEE Member # __________

Mailing Address __________

____

Country __________

Email __________

Phone __________

TCDE Mailing List

TCDE will occasionally email announcements, and other opportunities available for members. This mailing list will be used only for this purpose.

Membership Questions?

Xiaoyong Du
Key Laboratory of Data Engineering and Knowledge Engineering
Renmin University of China
Beijing 100872, China
duyong@ruc.edu.cn

TCDE Chair

Xiaofang Zhou
School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, QLD 4072, Australia
zxf@uq.edu.au

IEEE Computer Society
10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1314

Non-profit Org.
U.S. Postage
PAID
Los Alamitos, CA
Permit 1398