# Querying and Mining Heterogeneous Spatial, Social, and Temporal Data

Dissertation
an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität
München

eingereicht von
Maximilian Franzke

München, den 5. Dezember 2018

Erstgutachter: Prof. Matthias Renz
Zweitgutachter: Prof. Michael Gertz, Universität Heidelberg
Drittgutachter: Prof. Bernhard Seeger, Philipps-Universität Marburg
Tag der mündlichen Prüfung: 18. Juni 2019

# Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. 5.)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Franzke, Maximilian
_______________________________
Name, Vorname

München, 24.6.2019 Maximilian Franzke
____________________ ____________________
Ort, Datum Unterschrift Doktorand/in

# 1 Zusammenfassung

Während sich die allermeisten Aspekte des modernen Lebens durch die fortschreitende Digitalisierung verändern, vergrößert sich damit die Menge an Daten, die gemessen, gespeichert und verarbeitet wird, signifikant. Einerseits vergrößert sich die Datenmenge, andererseits erhöht sich auch die Komplexität der Daten selbst: Immer mehr Teilnehmer in sozialen und kollaborativen Netzen sorgen dafür, dass sich Daten schneller als je zuvor verändern. Obwohl unterschiedliche Arten von Geräten zwar Messungen der gleichen Datenmetrik vornehmen, tun sie dies aus ganz verschiedenen Gründen und in unterschiedlichen Kontexten – und eine Unzahl an Programmen und Apps kombinieren verschiedene Datenmetriken, um ihren Nutzern einen Mehrwert zu bieten. Dadurch haben Daten keine konsistente „Form" mehr, sondern existieren in verschiedenen Qualitätsstufen durch unterschiedliche Repräsentationen auf verteilten, inhomogenen Datenbanken und werden für jede Anfrage anders verarbeitet – auch während sie sich permanent verändern.

Diese Dissertation beschäftigt sich damit, wie man den Herausforderungen begegnen kann, die diese neue Art von Daten mit sich bringt: Wenn ein einzelnes Objekt in verschiedenen Domänen gemessen wird, existiert es nicht länger nur als einzelner Punkt in einem Datenraum, sondern es kann Repräsentationen in mehreren Räumen haben. Solch eine Multi-Repräsentation von Objekten erfordert neue Maßsysteme, Konzepte, Indexstrukturen und Algorithmen für Speicherung, Verwaltung und Anfragen. Während sich die Komplexität der Daten erhöht, explodieren zugleich die Kosten ihrer Verarbeitung: Die Beantwortung von Anfragen auf große heterogene Datenmengen sollte daher individuelle Eigenschaften der Daten berücksichtigen; und Data Mining hilft dabei, noch komplexere Abhängigkeiten und Beziehungen zu entdecken, was den Weg für weitere Anwendungen ebnet.

Im Rahmen dieser Dissertation werden mehrere neue Methoden und Lösungen zur Behandlung solcher multi-repräsentierten Objekte vorgestellt. Jeder Ansatz konzentriert sich dabei auf unterschiedliche Aspekte und ermöglicht effiziente Lösungen für diese Szenarien. Eine neue Indexstruktur für generische, multi-metrische Daten ermöglicht es, dynamisch gewichtete Ähnlichkeitsanfragen effizient zu beantworten und schlägt bisherige Vergleichsverfahren. Ein neuer Ansatz zur Beantwortung von Skyline-Anfragen im geo-sozialen Datenraum berücksichtigt domänenspezifische Eigenschaften und ermöglicht so eine effiziente Anfragebearbeitung, indem Berechnungen in der jeweils geeignetsten Metrik durchgeführt werden. Da soziale Daten neben der räumlichen, auch mit der zeitlichen Domäne verknüpft sein können, fokussiert sich eine vorgestellte Data Mining-Methode darauf, in einem sozialen Netzwerk einflussreiche Personen (sog. *Influencer*) zu finden, indem sie Interventionsanalysen auf temporal-sozialen Graphen durchführt. Außerdem werden in dieser Dissertation Methoden zur Anfragebearbeitung auf unsicheren räumlich-zeitlichen Daten sowie nutzergenerierten (*„Crowd-Sourcing"*) Graph-Daten vorgestellt.

# 2 Abstract

As most aspects of modern life face a continuing process of digitalization, the amount of data that is measured, stored, and processed increases significantly. While the volume of data itself increases, the complexity increases as well: With more and more participants of a social and collaborating web, data changes faster than ever before. Different types of devices measure the same metrics but within different contexts; and tremendous amounts of applications combine different data metrics to provide benefit to their users. Therefore, data does not have a consistent "shape" anymore but exists in several levels of quality in different representations in non-homogeneous distributed data bases and is processed differently for each query while it changes constantly.

This thesis focusses on how to approach the challenges of handling this new type of data: As the same entity is measured in different domains, it no longer exists as a single point in one data space but can have representations in multiple domain spaces. Such a multi-representation of objects requires new measures, concepts, index structures and algorithms to efficiently be stored, managed and queried. As the complexity of the data increases, the cost of working with it may simultaneously explode: Answering queries on large heterogeneous data sets should consider the data's individual properties and data mining approaches can discover even more complex connections and relationships, which opens the door for novel use cases.

Within the scope of this dissertation several new methods and solutions for handling such multi-represented objects are presented. Each approach focusses on different aspects and provides efficient solutions for complex settings. A new index structure for generic multi-metric data allows to answer dynamically weighted similarity searches and outperforms previous approaches. A novel approach to answer skyline queries in the geo-social domain considers domain-specific properties and allows for efficient query processing by performing calculations based on the most suitable metrics. As social data may not only be combined with a spatial, but a temporal domain as well, a new data mining technique to find influencers within a social network is presented that uses intervention analysis on temporal-social graphs. Furthermore, techniques on query processing on uncertain spatial-temporal data as well as crowdsourced graph data lie within the scope of this dissertation as well.

# 3 Contents

# 4 Thesis Details

This cumulative dissertation aggregates previously published research work, which consists of six contributions published at high-impact conferences within their research field. All publications were peer-reviewed by at least three program committee members.
During the years since 2014 the following contributions have been accepted and published at conferences within the computer science field. The bibliography chapter enlist all participating co-authors for every publication, while an additional section on impact factors gives an overview of the quality and impact of the venues the contributions were published.

## 4.1 Publications – Bibliography

Within this dissertation, publications will be referenced and cited through the "short-code" in bold font, which translates to the common venue abbreviation and year of publication. Please note that for the publications [DASFAA'14] and [SIGMOD'14] the department followed the chair's recommendation to enlist authors in alphabetical order.

**DASFAA'14** Tobias Emrich, Maximilian Franzke, Nikos Mamoulis, Matthias Renz, and Andreas Züfle.
**Geo-Social Skyline Queries**.
In: Sourav S. Bhowmick, Curtis E. Dyreson, Christian S. Jensen, Mong Li Lee, Agus Muliantara, and Bernhard Thalheim, editors,
*Proceedings of the 19th International Conference on Database Systems for Advanced Applications*, Part 2, volume 8422 of Lecture Notes in Computer Science, Bali, Indonesia, April 2014, pp. 77–91. Springer International Publishing.
ISBN: 978-3-319-05812-2, DOI: 10.1007/978-3-319-05813-9_6

**SIGMOD'14** Tobias Emrich, Maximilian Franzke, Hans-Peter Kriegel, Johannes Niedermayer, Matthias Renz, and Andreas Züfle.
**An Extendable Framework for Managing Uncertain Spatio-Temporal Data**.
In: Curtis E. Dyreson, Feifei Li, and M. Tamer Özsu, editors,
*Proceedings of the 2014 ACM International Conference on Management of Data*, Snowbird, UT, USA, June 2014, pp. 1087–1090. ACM, New York, NY, USA.
ISBN: 978-1-4503-2376-5, DOI: 10.1145/2588555.2594523

**ICDE'15** Gregor Jossé, Maximilian Franzke, Georgios Skoumas, Andreas Züfle, Mario A. Nascimento, and Matthias Renz.
**A Framework for Computation of Popular Paths from Crowdsourced Data**.
In: Johannes Gehrke, Wolfgang Lehner, Kyuseok Shim, Sang Kyun Cha, and Guy M. Lohman, editors,
*Proceedings of the 31st IEEE International Conference on Data Engineering*, Seoul, South Korea, April 2015, pp. 1428–1431. IEEE.
ISBN: 978-1-4799-7963-9, DOI: 10.1109/ICDE.2015.7113393

**ICDE'16** Maximilian Franzke, Thomas Emrich, Andreas Züfle, and Matthias Renz.
**Indexing Multi-Metric Data**.
In: Mei Hsu, Alfons Kemper, Timos Sellis, Boris Novikov, and Eljas Soisalon-Soininen, editors,
*Proceedings of the 32nd IEEE International Conference on Data Engineering*, Helsinki, Finland, May 2016, pp. 1122–1133. IEEE.
ISBN: 978-1-5090-2019-5, DOI: 10.1109/ICDE.2016.7498318

**ADC'16** Maximilian Franzke, Janina Bleicher, and Andreas Züfle.
**Finding Influencers in Temporal Social Networks Using Intervention Analysis**.
In: Muhammad Aamir Cheema, Wenjie Zhang, and Lijun Chang, editors,
*Proceedings of the 27th Australasian Database Conference*, volume 9877 of Lecture Notes in Computer Science, Sydney, Australia, September 2016, pp. 3–16. Springer International Publishing.
ISBN: 978-3-319-46921-8, DOI: 10.1007/978-3-319-46922-5_1

**EDBT'18** Maximilian Franzke, Tobias Emrich, Andreas Züfle, and Matthias Renz.
**Pattern Search in Temporal Social Networks.**
In: Michael Böhlen, Reinhard Pichler, Norman May, Erhard Rahm, Shan-Hung Wu, Katja Hose, editors,
*Proceedings of the 21st International Conference on Extending Database Technology*, Vienna, Austria, March 2018, pp. 289–300. OpenProceedings.org.
ISBN: 978-3-89318-078-3, DOI: 10.5441/002/edbt.2018.26

## 4.2 Impact Factor

While it is generally challenging to find a universal metric to rank each conference's or journal's impact within its field, there exist some approaches to rank conferences directly or categorize them into ranking clusters. The following table maps enlists various ranking scores for relevant conferences.

| | DASFAA | SIGMOD | ICDE | ADC | EDBT |
|---|---|---|---|---|---|
| **CS RANK**[1] | 0.79 | 0.99 | 0.98 | 0.75 | 0.88 |
| **CORE RANK**[2] | B<br>"good conference, and well regarded in a discipline area" | A*<br>"flagship conference, a leading venue in a discipline area" | A*<br>"flagship conference, a leading venue in a discipline area" | Australasian<br>"A conference for which the audience is primarily Australians and New Zealanders" | A<br>"excellent conference, and highly respected in a discipline area" |
| **ERA RANK**[3] | A | A | A | B | A |
| **MICROSOFT ACADEMIC**[4] | 28 of 263 | 2 of 263 | 3 of 263 | 36 of 263 | 8 of 263 |

## 4.3 Declaration of Contributions as Co-Author

### 4.3.1 DASFAA'14

Through my contribution of splitting the Bookmark Coloring Algorithm into incremental parts that can be calculated on demand, I laid the main foundation for our core algorithm submitted through the paper. By also proving that an additional upper bound for any node in the graph can be derived from the node's distance to the query node, the algorithm became even more efficient.

My design, implementation and evaluation of the experiments demonstrated that our proposed solution solves the geo-social skyline problem efficiently.

[1] CS Conference Ranking: http://perso.crans.org/~genest/conf.html

[2] CORE2017 Ranking: http://www.core.edu.au/conference-portal

[3] ERA2010 Ranking: http://www.conferenceranks.com/data/era2010_conference_list.pdf

[4] Microsoft Academic Ranking in "Databases": https://web.archive.org/web/20160420155441/http://academic.research.microsoft.com:80/RankList?entitytype=3&topDomainID=2&subDomainID=18&last=0&start=1&end=100

### 4.3.2 SIGMOD'14

Since the proposed framework completes a series of works on spatio-temporal uncertain data, some parts of the underlying query processor and data structures have been developed previously. As my contribution within this publication, I designed a set of efficient client interfaces that allow a stand-alone graphical user interface to connect to the database. With these interfaces, it is possible to submit queries on the one hand, and on the other retrieve query results and database contents in a suitable structure so that the client can handle the data volumes sufficiently.

I designed and implemented the graphical user interface (GUI) components, which contains sophisticated approaches to visualize and interact with uncertain spatio-temporal data intuitively and precisely. A ‘playback’ function allows to fluently browse temporal data within the state space as well as the geometric space – possible through constant optimizations for spatio-temporal data.

### 4.3.3 ICDE'15

With regards to data mining I contributed the idea of deriving the popularity of POIs by counting photos taken nearby; assuming people tend to photograph ‘scenic’ points of interest more likely.

Traditional way-finding algorithms for graphs aim at finding the most cost-efficient path. Through my contribution of a concept of how to model gain as cost, we could use efficient and established algorithms to answer our queries.

Finally, I implemented a visual GUI to interact with the framework and to submit and evaluate queries.

### 4.3.4 ICDE'16

As the main contribution of our paper, I designed the RR*-Tree and a corresponding similarity query that benefits from the underlying concepts. This includes a variable set and amount of reference objects per indexed metric.

Based on the ELKI-framework[5], I developed an implementation of our proposed PM-Tree and RR*-Tree index structures. This implementation was the foundation for my conduction of an extensive set of experiments which pinpoint the influence of numerous variables and measure query costs. The adjoining discussion showcases the benefits of each proposed solution in various scenarios. Performing the experiments included the generation of suitable synthetic data as well as using real world data (here: Twitter).

[5] ELKI: Environment for Developing KDD-Applications Supported by Index-Structures: https://elki-project.github.io

### 4.3.5 ADC'16

I developed the concept of generalizing the problem of influencer detection in social networks by explicitly considering the temporal domain of temporal social networks. This included the first known definition of influence in time-dependant attributed graphs by introducing the Social Influencer Score. The Top-k Influencer Query was presented along as well, which ranks individuals according to their score.
Since the Social Influencer Score requires the definition of a performance function, suitable implementations for these functions had to be found for the practical applications (here: finding influencers in collaboration graphs) and it was necessary to prove they are reliable and fitting. The idea of using Auto-Regression Integrated Moving Average models from the field of Intervention Analysis was brought in by myself; while Dr. Züfle gave advise on setting up the Hero-Experiment to verify the claim that the performance function is a suitable model for influence. I performed the conducting of the experiment based on real-world data which then showed that the ARIMA-based approach is valid for approximating Social Influencer Scores.

### 4.3.6 EDBT'18

Researching a valid baseline approach and developing a concept of applying it to the domain of temporal social graphs was done by myself. It consists of using Ullmann's backtracking algorithm for the subgraph isomorphism problem and extending it by adding more processing steps to furthermore consider the temporal domain. During the integration work I discovered two filter steps that could be applied along Ullmann's filters during query processing. These new filters are specific to temporal graphs and can therefore be applied in the context of this work's problem definition:

- As adapting Ullmann's algorithm to the temporal domain requires to perform the isomorphism search for every timepoint of the time domain, the query complexity increases linearly with the size of the time domain. This can be countered by applying a **filter based on the time offset**: At any timepoint, the actual graph may contain only a (very small in practise) subset of edges. Performing the isomorphism search on this smaller graph reduces runtime complexity drastically.
- Our problem focuses on temporal social networks, where the social graph is usually sparse, and the query pattern is way smaller than the total graph. This justifies running a **pruning filter based on network distance**: Calculating the maximum number of hops from a central node to all nodes in the query subgraph allows to limit the number of candidates to be refined. This filter can be applied to non-temporal subgraph isomorphism as well but is highly beneficial in graphs of similar kind as social networks.

Additionally, I contributed the procedure for performing the refinement work during the index-based query processing: After a simple subgraph structure from the query graph has been identified in the main graph through the index, a refinement is necessary to validate if other parts of the query graph are isomorph to the found candidate subgraph as well. Concepts from the baseline approach can be applied here as well, and a dedicated focus on assignment permutations helps to narrow the number of possible candidates.

During the research phase I developed and implemented a foundation framework to run and measure the proposed methods and gather quantitative and qualitative results for the baseline approach, index structures and the query processing. This includes designing conducting the experiments to highlight strengths and weak points of the approaches in typical and edge case scenarios. Similar to that, designing a compact, yet meaningful and significant running example to highlight the critical points at every phase of the query processing fell under my responsibilities and is included in the paper.

# 5 Thesis Summary

## 5.1 Introduction

The continuously advancing digitalization now affects a large portion of industries. While new products and technologies emerge, also internal business processes are being analysed, formalized, and automated wherever possible. Among simple control and monitoring processes, more advanced adaptive systems are being put into production, which are often advertised as “smart” services. These services are not an enclosed isolated system anymore with just a few defined input and output operations but are able to consider large and complex amounts of data for their calculations. In addition to the increasing amount of data, the data itself is becoming more complex: Data models are composed from multiple components, which are distributed amongst very different systems. Here, objects may have a representation in several databases, while the representations may vary strongly with regards to quality, timeliness, accuracy and considered aspect. While in theory, an “omniscient oracle” (which may be simply a monolithic central database) can find ideal results for queries through its permanent access to all data in every quality required; practical smart services in the real world must try their best to answer queries under various restrictions. Examples for such types of restrictions are:

- Access to an external data source may be disabled, restricted or limited. Social networks for example provide access to parts of their data to developers of third-party applications, but the number of read operations to this data is typically limited to protect their own business model. The scope of access given to third parties may change over time as well, e.g. Facebook restricting developer access to their API in the aftermath of the uncovering of Cambridge Analytica’s activities in 2018[6].
- Although several data providers measure and manage the same data attribute, it may be provided in varying quality, if the data was mined for different purposes. As an example, geographical data is available on a global scale through map services; but this data is much less precise in comparison to blueprints designed by an architect for a construction project.
- While one account may have full access to a certain database, it is possible that this dataset is being anonymized or required to be alienated for other; e.g. due to applicable jurisdiction for protection of privacy or competition concerns, like for

[6] Facebook Developer News: API and other product platform changes: https://developers.facebook.com/blog/post/2018/04/04/facebook-api-platform-product-changes/

example through the introduction of the *General Data Protection Regulation* in the European Union in 2018[7].

- Personal devices such as smartphones, health tracking wristbands and home automation technologies are able to collect, store and evaluate data in real-time; while remote services and analog processes such like surveillance satellites or the residents' registration office collect and update their data in much larger periods of time.
- The possibility to create and edit user-generated content, which dominates the so-called *Web 2.0*, causes ever-increasing pace of updates of databases' contents. Additionally, smart services that adapt themselves to the behavior of their users no longer perform stateless queries without context: Every query affects the results of future queries through feedback processes – using a list of best-selling products based on past sales for product recommendations will affect future sales and will have an impact on itself over time.

When these new types of services and applications want to answer the queries of their users, they must find ways and means to calculate correct results despite those restrictions.

## 5.2 Problem Settings

Basic concepts for query handling usually assume total access to a single database, in which datasets must be retrieved as efficiently as possible, which fulfill one or several criteria. If the type of query is known, a special query algorithm may for example be used, or an index structure can be constructed or the database itself may be pre-sorted according to some criteria. However, if no ideal access to the data is possible – for example due to one of the restrictions enlisted before – the cost for performing the query may increase drastically, when the search space can therefore not be pruned efficiently. If the query affects more than just a single database, e.g. when a *JOIN* operation must be performed, the cost for the query computation can increase significantly in practical applications when generic approaches for query handling are used. However, if a detailed analysis of the employed data sources and query types is performed, those generic query strategies can be improved noticeably by consideration of application-specific datatypes.

[7] Data protection in the EU: https://ec.europa.eu/info/law/law-topic/data-protection_en

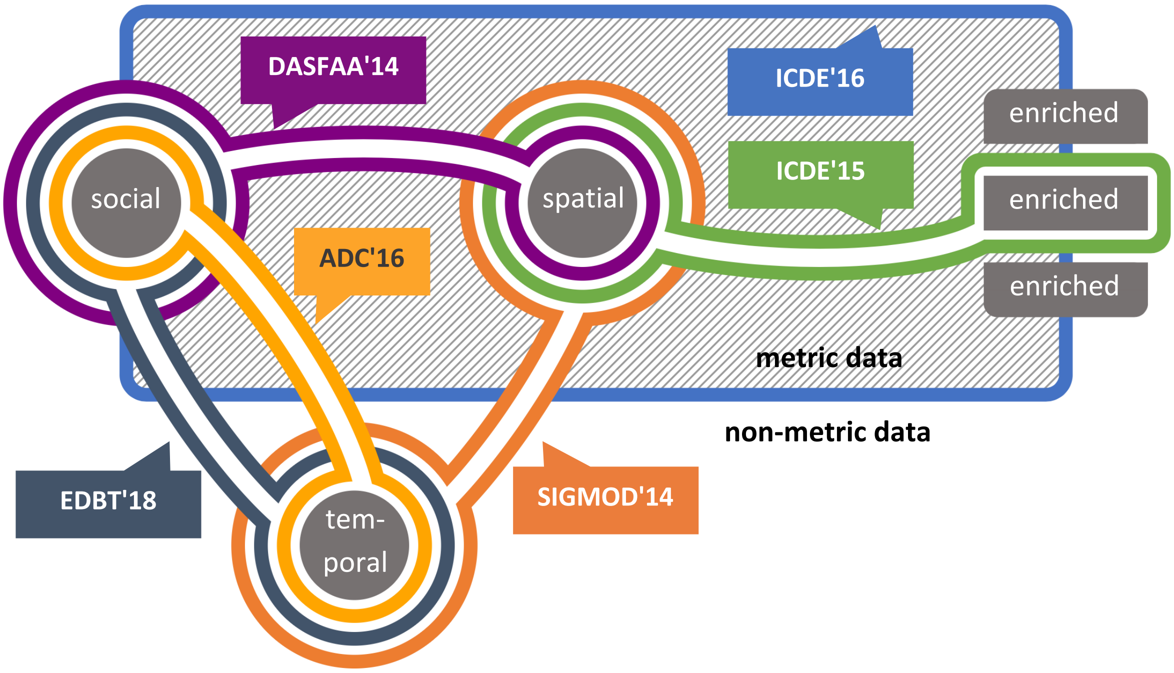


*Figure 1: Schematic illustration of how the published work covers various heterogeneous combinations of spatial, social, and temporal data.*

As this thesis focuses on heterogeneous data based on the spatial, temporal, and social domain, various combinations of these domains need to be treated. This thesis is not based on incremental contributions, but rather explores different problem settings of heterogeneous data exploratively. These problem settings will be explained in more detail in the following sub-chapters, while Figure 1 illustrates graphically how the published papers cover these different data domains.

### 5.2.1 Querying and Indexing Multi-Represented Heterogeneous Data Considering Dynamic Weights

Combining social networks with spatial information about the entities in the social graph enables novel application cases on the one hand, but on the other faces practical issues when executing queries which now need to consider two data spaces simultaneously. A special, yet relevant to real-world application cases, type of query for objects that can have a multi-attribute relationship with each other is the skyline query [7]. An explicit feature of skyline queries is that it is up to user to decide for a trade-off between the costs in different data spaces, so that no assumptions about the weights of each data space can be made in advance; however skylines help with narrowing down a range of choices for the user [8] [9]. For efficient computation of *Geo-Social Skyline Queries* the published approach in [DASFAA'14] considers the different costs of performing an accurate distance calculation in geometric space and in social networks, respectively: While a naïve method for skyline computation would determine both measures of distance for a candidate object, the proposed method only uses an exact calculation of the spatial distance, which can be achieved rather cheaply in comparison. For distances in the social graph however, only approximated values are used while still ensuring overall correctness of the results. These are either derived from already computed information or are iteratively refined until their exactness is suitable to decide whether they fulfill a certain condition, without spending effort to determine the exact value. Because meaningful measures of distance in social networks are expensive to compute, a vital increase in efficiency during query computation can be achieved by using this method.

The approach of supporting varying query weights across different data metrics provided by the user while simultaneously introducing an index structure that helps with reducing the total cost of processing the query is introduced in *Indexing Multi-Metric Data* [ICDE'16], while the method is extended to generic metrics and queries performance is increased by employing an index structure. Here a dedicated focus is put to the fact that in some metrics distance calculations are more expensive than in others (a generalization of the optimization approach of [DASFAA'14]) and retrieving exactly refined data rows from remote systems may incur costs as well. Two novel index structures are introduced with the *PM-Tree* and the *RR*-Tree* that can be adjusted and configured for varying data sources and metrics in a flexible manner.

### 5.2.2 Querying, Mining, and Indexing Temporal Social Networks

It is possible to model a social network through a graph, where for example nodes represent persons and edges express the friendship relations amongst them. Such a social network has been combined with spatial information in the previous chapter to answer *Geo-Social-Skyline-Queries*. However, if one were interested in not primarily spatial, but temporal aspects, new opportunities and challenges arise: With the knowledge about the historical evolution of a graph, more complex patterns and queries can be solved. In the publication [ADC'16] the *Top-k Influencer Query* was introduced, which evaluates historic connections between users to identify those that caused an impact on other users later on. This novel ranking query can be used for data mining, where the temporal information about a graph is used to mine a subset of nodes likely to have influential impact on others.

Adding the temporal dimension to a graph not only increases its complexity, but graph-related queries can be extended to consider temporal aspects as well. The subgraph isomorphism problem can be extended to not only find similar “structures” in the graph, but also “similar behaving” occurrences, like patterns of emerging friendship like *triadic closure*. The publication [EDBT'18] presents not only a method to extend existing subgraph isomorphism algorithms to the temporal domain, but furthermore presents an index structure capable of indexing a temporal social graph to allow for more efficient temporal subgraph matching.

### 5.2.3 Querying Uncertain Spatio-Temporal and Enriched Spatial Data

When considering the temporal along with the spatial domain, historical positions and trajectories of objects can be modelled analogously to the previously presented concept of the evolvement process of social networks. In practice however, a spatio-temporal database is more likely to be sparse or discretized at another level of granularity than the queries would require it to be. This means that the position of objects may not be known for every possible time point when observations are infrequent – the most important time point being "now", i.e. when no information about an object's current location is present. However, considering possible world approaches assumptions about an object's location between two observations can be made and [SIGMOD'14] presents a *Framework for Managing Uncertain Spatio-Temporal Data*. This framework allows to answer queries on the data probabilistically by involving techniques which adapt traditional concepts to uncertain spatio-temporal data [10] [11] [12] [13].

However, the challenge does not stop at uncertain data, which may not be present in the dataset in the desired level of granularity; it continues to applications where such data is not present at all. While a database may contain detailed geographical information about city and road maps, data in this form may not be needed for all application cases. The work [ICDE'15] realizes that it is more natural and comfortable for humans to memorize a route by important waypoints, or places of interest. This challenges the application to not only find navigation routes that are short and quick, but also easy to describe and remember. By introducing a query to find *k-Constrained Pareto Optimal Popular Paths*, the data needs to be enriched by the attribute *popularity* and the wayfinding algorithm will then optimize for paths that touch popular points of interest along their way. The popularity of points is hereby mined from crowdsourced data and result paths can be measured by the novel attribute *popularity* along traditional attributes like *distance* or *duration* in for example a skyline algorithm.

## 5.3 Skyline Queries on Geo-Social Data

### 5.3.1 Problem Motivation

Skyline queries are an important tool for answering user queries on heterogeneous data as they allow the user to weigh different dimensions even after query time, thus making it very generic and flexible. Personal preferences can be fed by the user and thus search criteria fit the individual user's demand.

The publication [DASFAA'14] focuses on skyline queries in the specific application field of spatial data in combination with social data and delivers the following contributions:

- Consideration and exploitation of the different properties of the underlying data dimension allow for fast query processing while not requiring an index structure for the social graph (which would require immense overhead for frequent insert, update and delete operations in the graph).

- Allowing for real-life relevant social distance function *Random-walk-with-Restart* and *Bookmark Coloring Algorithm* respectively, which do not even fulfill metric properties (triangular inequality). The functions depend on a *personalization factor* that can be set at query time and determines the balance between the importance of a node in the graph and the importance of a node in relation to a query node, thus offering a maximum level in personalization to the user.

### 5.3.2 Dataset

Real-world geo-social data may contain complex correlations and patterns which may not be replicated well through synthetic data generators. Therefore, the experiments for [DASFAA'14] were executed on real-world data only, namely the *Gowalla* dataset[8] [10]. Gowalla was a social networking site where users could virtually "check in" to locations they were currently at (similar to past *Foursquare*[9] or current *Swarm*[10] functionality). The dataset has been anonymized and consists of list of friendship links and a history of user check-ins into locations.

Geo-Social Skyline Queries are run on a social network, consisting of links between users (implying for example "friendship") and a database where each user is assigned a geographic location. Both locations and links are relevant only for the current point of time. While friendship links can be directly extracted from the Gowalla dataset, a location for every user for a certain time-point is not available. Therefore, such a database is artificially created from the users' history: Each user's location is defined as the location of their last check-in.

### 5.3.3 Approximating Social Similarity and Distance

A special focus is put on how the social distance in the graph is defined. As the *Random-walk-with-Restart* (RWR) [15] distance is an advanced flexible way of identifying relevant nodes in a graph with respect to a given node [16] it is used for the skyline computation. The *Bookmark-Coloring-Algorithm* (BCA) [13] is mathematically equivalent to RWR but calculated differently. While exact results are similarly hard to compute with BCA in comparison to RWR, BCA provides faster approximations of a distance, which can furthermore be limited through an upper and lower bound. These bounds are then used for the skyline computation, as it is sufficient to just determine *whether* an element is contained in the

[8] Stanford Network Analysis Project: Network Datasets: Gowalla: http://snap.stanford.edu/data/loc-gowalla.html

[9] Foursquare: https://foursquare.com/

[10] Swarm: https://www.swarmapp.com/

skyline, and not its exact distance properties to the query object. The paper [DASFAA'14] proposes two bounds: A rough one based on the network distance between a candidate and the query object, and a dynamic one, which continuously improves as iterative steps of the BCA are performed. A huge performance gain is achieved by only refining the bounds so much that a pruning or hit decision can be made, which in practice leads to profound boosts in performance.

### 5.3.4 Algorithm Performance

The proposed algorithm contains a further improved version as well, which is able to prune the complete remaining candidate set but requires accessing nodes according to their spatial distance to the query node – which requires spatial sorting at the beginning. Furthermore, two additional pruning steps may be added that allow to prune more candidates through the information gathered about other candidates:

- When a distinct candidate is refined, it may dominate other candidates, which can then be removed immediately.
- Candidates from the backlog can dominate each other. The refinement of a distinct candidate may refine other candidates as well, so an additional check whether new dominations occur within the backlog may prune additional candidates.

The later pruning check is foreseeable too expensive to be executed every time a refinement has happened. Therefore, it can be postponed to only be executed every $n$ refinements. Figure 2 illustrates the modularity of the filter steps.

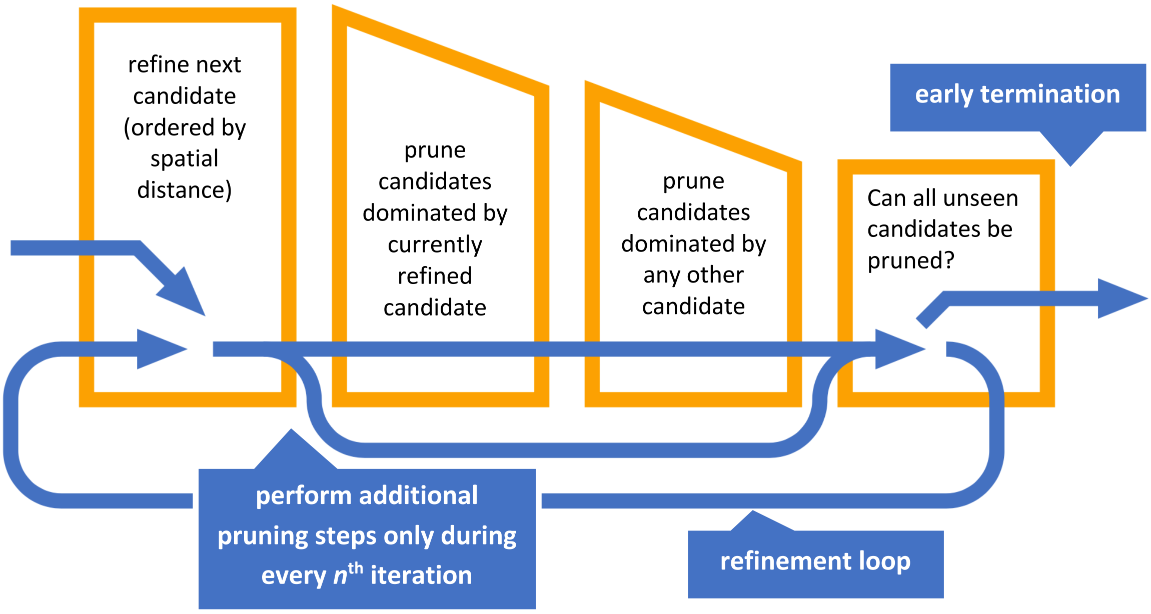

refine next candidate (ordered by spatial distance)
prune candidates dominated by currently refined candidate
prune candidates dominated by any other candidate
early termination
Can all unseen candidates be pruned?
perform additional pruning steps only during every $n^{th}$ iteration
refinement loop

*Figure 2: Modular pruning steps for the improved Geo-Social Skyline Query algorithm.*

Because the improved version of the algorithm requires the construction of a spatial index structure to allow distance-based browsing, and the optional pruning methods require comparison calculations as well, it is crucial to evaluate whether the payoff of the pruning potential justifies investing in the additional checks. Therefore, an experiment is conducted, where each version of the algorithm is compared against each other and measured by their overall runtime, thus demonstrating whether the precomputation steps balance faster pruning, and a filter step should be included in an efficient algorithm. To prove generic applicability, the experiments are conducted across the range of the personalization parameter $\alpha$ of RWR and BCA, which allows for a balance between nodes that are relevant to the query object and objects that are generally important in the network. Because $n$ controls how often additional pruning calculations are performed and therefore influences performance as well, both a value of $n = 1{,}000$ and $n = 10{,}000$ are considered. Since results show that neither of the additional dominance checks described results in a better runtime, no further experiments are necessary to evaluate the best setting for $n$, as it is outperformed by the version that just includes early termination functionality.

### 5.3.5 Qualitative Evaluation of the Problem Definition

As the *Geo-Social Skyline Query* was introduced in [DASFAA'14], it needs to be shown whether the problem definition brings any practical benefits instead of being a rather theoretical concept. Therefore, the result of the query is examined to indicate usefulness: The idea of a skyline is to give the user a small, however diverse set of pareto-optimal objects to choose from. If the set is too large to be presented to a user (i.e., thousands of search results) or too small (e.g., none or one object), it lacks practical use. By measuring the average size of the skyline result set through various settings this argument can be addressed. In various experiment settings involving sparse and dense geographic query points, well-connected and lonely query nodes in the social graph and varying personalization factor $\alpha$, the skyline size is measured to contain roughly fifteen objects across all settings, which is a very feasible amount of results and therefore shows the practical relevance of the problem definition.

### 5.3.6 Discussion

The proposed concept of geo-social skyline queries was proven to be relevant and applicable to real-world applications, as qualitative and quantitative evaluations on an actual dataset have shown.

In contrast to many other concepts, this approach allows to use the more realistic *Random-walk-with-Restart* distance as a measure for the social distance (instead of for example just the network distance), which is a more suitable approach for geo-social networks and commonly adapted in academia, e.g. [14] and [15]. However, the RWR-distance is expensive to compute, and while the proposed algorithms consider these costs and try to minimize them especially, these optimizations exploit specific properties of the *Bookmark Coloring Algorithm* (which is equal to RWR). This implies that the proposed skyline algorithm cannot be applied to other measures of social distance with ease.

## 5.4 Indexing Multi-Metric Data for Efficient Dynamic Similarity Search

### 5.4.1 Problem Motivation

When objects are represented in a heterogeneous dataset, users may query that data using constraints that span across all data dimensions. Range and window queries can be adapted to span all dimensions and by limiting the search space for each dimension, the user can narrow the size of the result set. However, for practical applications *similarity queries* are much more helpful, as they put the dataset in relation to a query object, thus *personalizing* it. Another important problem arises as soon as queries do not rely on absolute arrangements of the dataspace: Because data dimensions may be independent from each other, distances in these dimensions are as well. As soon as these independent distances have to be aggregated together, one has to face the question of how they should be weighed against each other. While a fixed decision for a certain application case may result in query performance increase, it limits the flexibility and freedom of the query. Thus, it is most practical to leave that decision up to the user, i.e. making no presumptions about query weight distribution until query time and allowing for *dynamic* similarity search. The *Multi-Metric Similarity Query* (MMSQ) allows for such flexibility, as it retrieves all objects $o$ from the database $\mathcal{D}$ which under consideration of the user-provided weight distribution $\omega$ have a multi-metric distance $\mathrm{m}^{\omega}$ to the query object $q$ shorter than a range $\varepsilon$:

$$\mathrm{MMSQ}(q, \omega, \varepsilon) \coloneqq \{o \in \mathcal{D} \mid \mathrm{m}^{\omega}(q, o) \leq \varepsilon\}$$

The concept of consideration for the different costs of distance functions in different data domains is generalized in [ICDE'16]. Here two novel generic index structures for multi-metric data are introduced that allow for efficient Multi-Metric Similarity Queries:

- The *PM-Tree* is a combined index structure which allows individual index structures for each metric. These structures need to support distance-based browsing of data and the PM-Tree includes a method to coordinate the browsing of the individual metrics and compute the final result set.
- The proposed *RR*-Tree* can adapt to different levels of "hardness" of metrics.
- Different levels of refinement account for potentially expensive cost of distance computations in modern applications.
- The user can decide at query time how to weigh the individual metrics against each other to retrieve an ordered list of relevant objects.

### 5.4.2 The PM-Tree and RR*-Tree

Instead of combining all metrics into a single index like the $M^2$-Tree [20] and $M^3$-Tree [21] do, the **PM-Tree** keeps an individual index for each metric. For the generic case this can be a basic M-Tree [17], but if specifics about the metrics are known, other suitable data index structures applicable to that metric and supporting the essential queries are may be used as well.

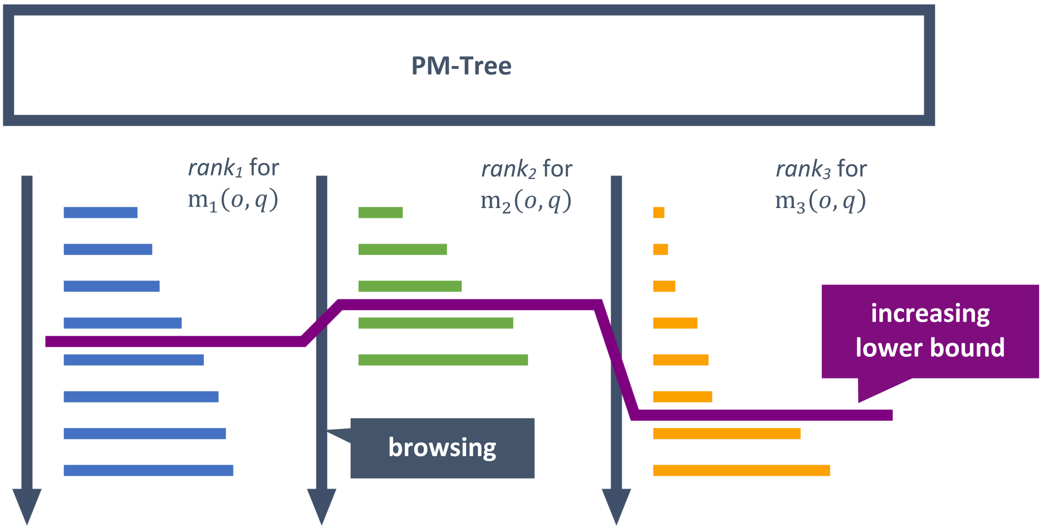

PM-Tree
$rank_1$ for $\mathrm{m}_1(o,q)$
$rank_2$ for $\mathrm{m}_2(o,q)$
$rank_3$ for $\mathrm{m}_3(o,q)$
increasing lower bound
browsing

*Figure 3: PM-Tree illustration for three metrics. In each metric, a rank is performed to order objects in relation to the query object. This continuously improves the lower bound for unseen or unrefined objects and enables pruning.*

To support Multi-Metric Similarity Queries, the PM-Tree considers the weights across the metrics the user has provided at query time. The PM-Tree performs a ranking query in each of its sub-indexes and retrieves objects from the indexes one-by-one and in parallel, which is illustrated in Figure 4. As the distances to the query object continuously increase, browsing can be terminated as soon as the query distance threshold is reached, as all further candidate objects can be safely pruned [23] [24]. As the user query weights may have a strong bias against one or some metrics, these metrics can be prioritized during the parallel refinement to reach the termination point faster.

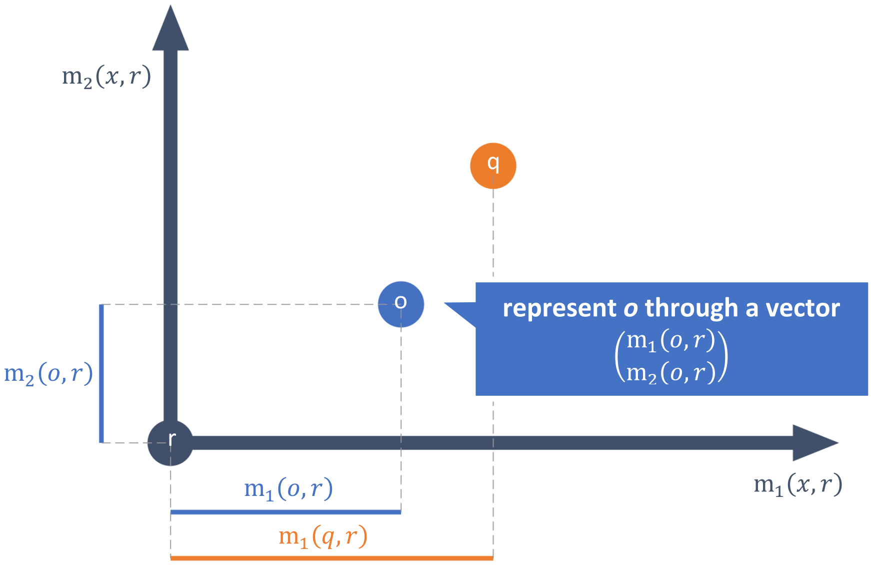

$\mathrm{m}_2(x,r)$
q
o
represent $o$ through a vector $\begin{pmatrix}\mathrm{m}_1(o,r)\\ \mathrm{m}_2(o,r)\end{pmatrix}$
$\mathrm{m}_2(o,r)$
r
$\mathrm{m}_1(o,r)$
$\mathrm{m}_1(x,r)$
$\mathrm{m}_1(q,r)$

*Figure 4: Exemplary representation of an object $o$ by its distance to a reference object $r$ in a two-metric scenario to visualize the concept of the RR*-Tree. By determining the distance of the query object $q$ to $r$ as well, the distance (or similarity) $m_i(q,o)$ can be bounded through the triangular inequality:* $|m_i(q,r) - m_i(o,r)| \leq m_i(q,o) \leq m_i(q,r) + m_i(o,r)$.

The ground laying idea of the **RR*-Tree** is the representation of the objects in another data space. By choosing a reference-object-space, each object is represented by its distance in each to a global reference object in each metric, as illustrated in Figure 4. Therefore, a vector can be a representation of the object and a traditional R*-Tree [25] can be used to index those vector representations. As it is assumed that similar objects will have similar distances to the reference objects, this increases the likelihood that similar objects will be indexed in the same data pages in the R*-Tree. For query evaluation, the query object's distance to the reference object will be computed at query time and using the triangular inequality (which holds true for metric spaces), the distance from the query to candidate objects can be estimated. The paper also introduces the possibility to choose different reference objects for different metrics and even to choose more than one reference object per metric to improve the lower and upper bounds yielded through the triangular inequality.

### 5.4.3 Performance Evaluation of the Proposed Index Structures

As the publication proposes two novel index structures to handle similarity queries on multi-enriched objects with an arbitrary weighting function between the metrics provided just at query time – namely the PM-Tree and the RR*-Tree – both new methods need to be evaluated in comparison:

- against each other to provide an explanation which indexing method is beneficial in which conditions

- with an existing method: The M$^3$-Tree [21]
- against trivial methods to prove that the problem is so complicated that it cannot be treated efficiently

While the M$^3$-Tree, the RR*-Tree, and the PM-Tree each have fundamental underlying concepts, their performance may be affected by different configuration parameters, attributes of the handled data, and parameters of the similarity query. To either prove the superiority of one of the three approaches or explain the scenarios where one is advantageous to another, all those variables and parameters that can affect the cost of the query need to be identified and evaluated:

- the number of objects inside the database
- the number of attributes or metrics the objects are represented in
- the "hardness" (internal complexity) of each available metric
- the relative weight distribution across the metrics provided at query time
- *specific to the RR*-Tree:* The number of reference objects used during index construction
- *specific to the RR*-Tree:* The distribution of all available reference objects across the individual metrics

Another important aspect for performance evaluation is which cost should be accounted for. Traditional research methodology mainly focusses on computation time and disk or RAM space. However, in times of distributed computation platforms and use cases, a more detailed analysis seems appropriate for this work. For example, in practice the cost of comparing two objects by their attributes through a comparison function requires on the one hand the calculation of a formula (a moderately easy task), but also prerequisites retrieving the attributes beforehand. In a world where an object may be represented in different services and databases (and not a central core storage), this task quickly becomes expensive. Therefore, it is obvious that the total count of similarity calculations (or distance calculations) can affect the total runtime significantly. Thus, the experiments will account for the total number of distance calculations separately. By denoting the count in absolute numbers, the conversion into practical runtime can then be achieved later; when for example different calculation times per use case (locally, remote, or distributed) can be applied.

### 5.4.4 Evaluating Influential Parameters

As explained before, various parameters have been identified that may have an impact on query performance. Since each parameter has its unique domain of valid configuration values, the total count of all possible attribute combinations is unfeasible. Because of this impediment, and secondly the interest to examine each parameter's individual influence on the query scenario, the following general experiment design was chosen: A default scenario was defined, where for each of the listed parameters has a default value. Then a dedicated experiment was run for each parameter in which only the current parameter was varied across its domain. This allows to see an individual parameter's role in the query phase and limits the number of experiments to perform drastically. However, complex cross-correlations of parameters cannot be detected through this setup. In practice this seemed nonetheless not to be a problem, as each parameters influence could clearly be described and did not follow arbitrary patterns, thus allowing for the conclusion that the parameters are independent from each other.

The range of the domain from which a parameter's value can be from was chosen by design and then iteratively extended until a clear trend could be the experiments could be observed.

Furthermore, the underlying data has been varied as well. To show practicability in real-use applications, a Twitter dataset was chosen, which contains 100,000 different tweets. Five different metrics of those tweets have been chosen to be used in the experiment:

- The text content of the tweet and by definition the distance between two texts was chosen to be equal to the Levenshtein-distance [26].
- The length of the tweet in count of characters. As this metric is obviously highly correlated with the previous metric, it is used to model correlation between metrics in the experiments.
- The geographic reference of a tweet to measure the geographic distance between two tweets. Classical latitude/longitude coordinates have been transformed to $(x|y|z)$ coordinates to allow the use of the Euclidian distance function, which in contrast to the orthodrome-based distance function satisfies the triangular inequality (thus making it a metric).
- The friend count of the author of a tweet; thus, classifying tweets from popular influential people.
- The hue value of the background color the author of the tweet has chosen for their profile page. This may seem rather random, and it is meant to be. While no correlation is expected between this metric and the others, this metric's influence on the results can be compared to the tweet length metric, which on the contrary is highly correlated.

Additionally, experiments have been run on a synthetic dataset as well. While "good" synthetic datasets are hard to produce, they allow to manipulate underlying properties of the dataset precisely and therefore allow a more quantitative analysis. In the context of this paper, synthetic datasets provide the possibility to set the *intrinsic dimensionality* of a metric, i.e. its hardness. This concept can be visualized as follows: Assuming an object’s spatial location is described by a single value (e.g. its distance to a special point of interest):

- In a one-dimensional space, like for example a sprint on a racetrack, a runner’s distance from the starting point describes their position in geometric space perfectly. Similarly, two runners with an equal distance from the start point are at the identical position.
- In a three-dimensional space, a planet’s distance to the sun limits its spatial location drastically to all points on an imaginal sphere around the sun. However, we cannot make assumptions about the spatial distance between two planets with similar distance to the sun. They may either be really close in three-dimensional space, or pretty far away, being located at opposite points of the sun.
- In a ten-dimensional space, it becomes even more difficult to make statements about individual object’s properties. If a student’s average grade is a “C”, it becomes nearly impossible to correctly assume their individual grades of the courses they took.

### 5.4.5 Discussion

The evaluation of the proposed PM-Tree and RR*-Tree have them proven to be viable solutions and to be more efficient than traditional approaches, especially when separately accounting for page accesses, indexing overhead and distance calculations. These performance measures are more relevant today, as they are flexible enough to account for remote database lookups and computation of very complex distance functions, like social distance. The novel index structures are heavily optimized towards these criteria and thus support handling of (partially) unrefined objects.

The PM-Tree can outperform the RR*-Tree in scenarios where there is a larger imbalance in the weight distribution among the metrics, while the RR*-Tree allows to combine “easy” and “hard” metrics in a single, performant index structure.

However, the *Curse of Dimensionality* limits the level at which the RR*-Tree can be optimized [25]: As each reference object chosen for the metrics increases the dimensionality of the internal R*-Tree, there is a practical limit of reference objects to be used and the challenge shifts to how the available reference objects should be distributed among the metrics to gain the best increase in performance. The paper proposes a heuristic approach of how to achieve this, by estimating how many more nodes may be pruned using an additional reference object.

## 5.5 Data Mining in Temporal Social Graphs: Identifying Influencers

### 5.5.1 Problem Motivation

In a temporal social graph both nodes and edges may undergo changes as time continues: Edges can be created, removed, intensified and diminished; whereas the nodes themselves may change as well, as they might have associated attributes which can vary over time. *Influencers* in a social graph are people that have an effect on others, like increasing the popularity of a brand with peers or enabling friends to get better grades in study classes. While the effect of the influence may have a temporal delay, it becomes challenging to identify the nodes which have caused the impact in the past to honor their efforts.

The paper [ADC'16] proposes two new contributions: First, the definition of *influence* in time-dependent attributed graphs and followed by the definition of the *Top-k Influencer Query*. Second, simple performance functions and more advanced performance function based on intervention analysis [26] [27] to model and approximate the change in performance over time through influence are presented as well.
Since the problem definition itself is new, it has to be shown that the results are viable as well as giving different insights than other problem definitions (qualitative evaluation). The different performance function should be compared against each other to give answers about the usefulness, quality and performance of their results (quantitative evaluation).

### 5.5.2 Dataset

To prove the problem's usefulness in real-world applications, it is essential to analyze it on an authentic dataset. The ACM citation graph is a viable instance of such a dataset: Discretizing it to a yearly interval smoothens out seasonal fluctuation but keeps the time domain large enough to observe temporal changes. It is furthermore a dataset close enough to the main motivation: While influencers can also occur in social networks and advertising, where the influenced domain is sales or brand value; influence in the research field is also obvious. For practicality, the dataset has been reduced to the set of people who have had at least one publication at a SIGMOD conference (hence, the database community).
Since there needs to be one or more attributes to be influenced, a person's *publication count* is chosen as the key performance attribute for various reasons:

- It is definitely and easily measurable
- Publication count is essential for PhD students as well as post-docs and professors (at least in the computer science field from which the ACM dataset is from)
- The popular and widely respected h-Index of a researcher is accepted as a viable indication of a person's research in the academic field

As with [DASFAA'14], no efforts were made to artificially replicate a synthetic temporal social graph, as dependencies in the social and temporal domains are too complex for a data generator and infer the risk of bringing an unwanted bias. However, the ACM dataset will be punctually mutilated in dedicated experiments explained later.

### 5.5.3 Modelling and Measuring Influence

To formalize the term *influence*, performance functions are introduced. They can be applied to a node $v$ in the graph under consideration of a specified attribute $a$ this node has. Considering a time point $t$, the performance function expresses the quality of an external intervention on $v$ at time $t$. The proposed performance functions only depend on available data, namely the time-series $a(v)(t)$ and aim at spotting the exact moment (or period) the intervention was happening. Generally speaking, performance functions seek for a change in the attribute's value under consideration that an impact may be delayed. Two concrete performance functions are introduced:

- **Before and After Average** compares the average values of the time series in a defined window before respectively after the examined timepoint. This simple method aims at finding direct changes in the timeseries.
- **Auto-Regression Integrated Moving Average (ARIMA)** is a stochastic intervention model aimed at modelling the impact on a timeseries using maximum likelihood estimation [28] [29].

As each node's performance can now be expressed through a performance function, the graph structure can now be used to calculate a *Social Influencer Score* for each vertex: For any timepoint, the influencer score of a node is increased by the value of the performance function of the nodes the node is currently connected with. This directly attributes a node's performance to its possible causes. The *Top-k Influencer Query* now identifies the set of $k$ nodes with the highest Social Influencer Score.

### 5.5.4 Evaluation

After performing the *Top-20 Influencer Query* on the ACM dataset, the resulting top influential researchers are manually evaluated and compared to other renowned ranking methods (Microsoft's field ranking score and the h-index retrieved through Google Scholar). The comparison shows that the *Top-k Influencer Query* does not give an arbitrary, however also not a correlating ranking to the other scoring methods. It furthermore seems to indicate that it indeed is a new, viable method for finding influential researchers.

However, it is yet to be proven that the ranking itself is correct on the inside – as for example the *random* ranking would be different to established ranking approaches as well. To show that the new query actually finds influential people, the so-called *Hero* experiment was designed. Its key idea is to create a single influencer (the hero) artificially and then verify that the *Top-k Influencer Query* identifies them correctly. For a single influencer to exist, the rest of the graph must be completely random, e.g. a synthetic network. The network is then adjusted that the nodes adjacent to the hero will experience a performance boost over the time period of two time points after the interaction with the hero. For this scenario, generating a simple synthetic network is applicable, as no qualitative or quantitative analysis on the network or the query results is being done – only the *hero* is being evaluated. Repeatedly executing this experiment shows that the hero's position rises in the ranking, and the higher the performance boost given the more their position in the ranking climbs to the bottom. On the one hand, this obviously shows that the influencer is clearly identified through the proposed method. On the other hand, the experiment also verifies that the length of the influence can be nontransparent to the ranking algorithm, as it is nevertheless able to find the best influencer.

As the problem definition of the Social Influencer Score relies on the existence of a performance function intended to model the quantity of an impact on a node at a given time, two performance functions are presented in the paper: The sophisticated ARIMA-function (Auto-Regression Integrated Moving Average) based in intervention analysis in the field of statistics, along with the the Before and After Average Function (BaAA) that takes the average of a few points before the time point ant the average of a few time points after, and compares these two values. For obvious reasons, the later performance function is much easier to compute than the sophisticated ARIMA-model, for which solvers exist. This is supported through quantitative results: The calculation of single performance function value using ARIMA is approximately five orders of magnitude more expensive than BaAA. However, BaAA requires the specification of the number of time points to consider; i.e. an input variable that has to be provided with domain knowledge. Knowing (or guessing) the correct number can help BaAA to achieve equally as good results as ARIMA (verified by the *Hero* experiment, where the number can be set to the same value as the duration of the artificial influence as explained earlier), however with an arbitrary window size the quality usually lies beneath that of ARIMA. Therefore, a clear trade-off between optimal results and speed performance can be provided for application-specific use.

### 5.5.5 Discussion

Correctly identifying and attributing the influencers in a social graph is an important classification challenge, as it looks beyond graph structures and considers attribute values and their temporal change as well.

Using the statistical ARIMA-model for a performance function identifies those influential nodes, however performing the maximum likelihood estimation necessary is very expensive as it requires tremendous computation time as it needs to be performed for every node in the graph and every timepoint in the time domain. However, with the much simpler Before-and-After-Average performance function a much more efficient, yet still accurately enough alternative is presented.

The scope of this publication does not yet consider that even an influencers score may vary over time – i.e. they might evolve from a bad to a good influencer. Considering this fact in future research work would require adjustments to the query definition, depending on whether one is searching for the all-time best influencers or the best influencers right now. The later question can easily be answered by artificially truncating the time domain.

## 5.6 Pattern Search in Temporal Social Graphs

### 5.6.1 Problem Motivation

In the previous section it was shown how to identify individual nodes that influence attributes of other peers in a temporal social graph. However, one may not only just be interested in individual nodes but may be seeking for patterns of structural and temporal shape in the graph, like when a group of friends is formed. These patterns can be searched for by using *Temporal Subgraph Matching* presented in [EDBT'18]. Informally, we are specifying a structural graph and define its temporal behavior, i.e. edges evolving, disappearing etc. Now all subgraphs of in the social graph should be identified where the nodes behave in the same temporal pattern as specified in the query.

Subgraph isomorphism problems are in general NP-hard. To make the problem solvable with reasonable resources, it must be examined whether heuristics can be applied in the individual fields of application. As this work focusses on temporal social networks, such a limitation on the general problem is the case.

### 5.6.2 Dataset

All experiments are run on a derived dataset (called *PUBS*), which is based on the ACM citation graph used in [ADC'16] as well. For PUBS, the nodes from the citation graph (namely the researchers) are extracted and the time-domain is discretized into yearly intervals. If two researchers have published a work together included in the ACM citation graph, an edge between them is created in PUBS for the year of the publication. PUBS can therefore be described as a collaboration graph, and the focus on annual intervals is a viable and meaningful discretization of the time-domain, as it evens out seasonal factors and is still precise enough to allow for change during a researcher’s career. The evaluation of PUBS already shows that it is relatively sparse, which is beneficial to the solution of the subgraph isomorphism problem.

### 5.6.3 Temporal Subgraph Matching

The publication demonstrates, how a traditional algorithm by Ullmann [18] for the subgraph isomorphism problem can be extended to consider the temporal domain as well. This is achieved by first projecting the graph und query graph along the temporal domain (i.e., creating a non-temporal social graph), solving the subgraph isomorphism query, and then reintroducing the temporal aspects and performing a last refinement step to account for the temporal correctness as well. Here two new filter steps are introduced for Ullmann's subgraph isomorphism algorithm:

- **Pruning based on the time-offset** only considers parts of the graph relevant for a specified time. This may result in a much sparser non-temporal graph on which the subgraph isomorphism is performed, hence an increase in performance.
- **Pruning based on network distance** discards all parts of the graph whose network distance is greater than in the query graph when generating assignment mappings. This reduces the number of candidates to be checked and is also a generic application for non-temporal graphs.

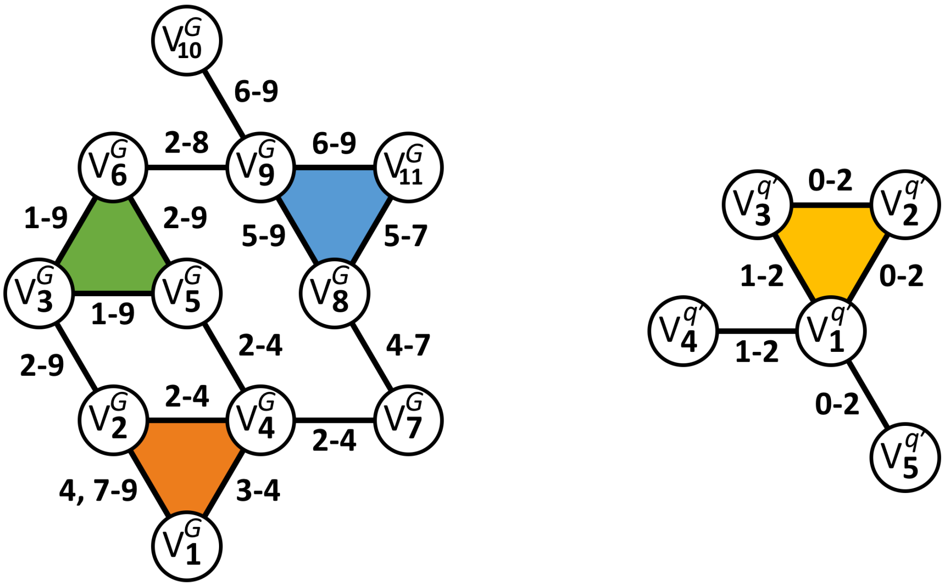


*Figure 5: Exemplary temporal social graph (left) and temporal subgraph query (right). Numbers at the edges indicate, at which time points a connection between the vertices exist. Choosing a "triangle" (coloured in the picture) as a Simple Subgraph Structure helps to find candidate locations in the graph.*

To further improve the process of mining candidate subgraphs, an index structure for temporal social graphs is presented, which allows to index the temporal behaviors of specified *Simple Subgraph Structures* (SSG) by encoding the individual states at a certain time through symbols. These symbols can now be concatenated to represent the temporal behavior of a simple subgraph occurrence. All occurrence’s encodings can now be stored in an index structure that supports efficient substring search. During query time, an SSG can now be identified in the query graph and its matching occurrences in the index can then be retrieved efficiently – illustrated in Figure 5. Additional components to the SSG in the query graph require an afterwards refinement.

### 5.6.4 Evaluation

As the main idea of the proposed index-supported search relies on the use of Simple Subgraph Structures, a list of basic SSGs is presented. All occurrences of the SSGs are then mined in the dataset PUBS and the count gives a foundation to discuss the question which SSG is a suitable selection for index construction, as the general idea is that the index works better if the SSG itself is more complex and specific, while a simpler SSG has a higher probability of occurring in the query graph at all. Evaluation shows that very simple SSGs without "loops" (I.e. chains of edges) occur quite frequently in a social graph, while more complex structures based around the "clique" structure (I.e., all or mostly all vertices are connected in the SSG) are scarce. Note however, that due to the internal symmetry of a clique an SSG has exponentially more permutations occurring in the graph the bigger the clique becomes.

Besides SSGs, typical query graph structures need to be evaluated as well to give feedback about the feasibility of temporal subgraph matching. Various query graphs consisting of up to five nodes are then enlisted – each containing a triangular substructure. Based on this set of query structures, it can be analyzed how useful different SSGs are in regard to query performance. An SSG is useful, if it does not produce unnecessary candidates. More generic SSGs (like a *string*) deliver more matches in the graph than complex SSGs (like a *triangle*). The difference in the experiments is about three orders of magnitude. However, if an SSG is not contained in the query graph at all, it becomes useless all together. Hence, the evaluation helps with finding a well-balanced SSG suitable for efficient search.

For examining the influence of the temporal length of the query on calculation time and size of the result, we generate a random temporal pattern for the set of SSGs mentioned before. In order to retrieve meaningful results; the pattern must in practice be not completely random but have at least one occurrence in the graph (which can be picked by random). Starting with a very *long* temporal pattern, and then iteratively reducing the temporal length consecutively, we can observe how more and more other parts of the graph become isomorph to the shorter query pattern. The number of results as well as the runtime of the query both increase exponentially, the smaller the temporal length gets. This leads to the conclusion that temporal subgraph isomorphism can be performed efficiently, when the temporal part of the query contains some amount of complexity (and thus uniqueness).

In direct comparison, the index-based approach is faster than the baseline approach by orders of magnitude. When the temporal length of the query increases, the performance gain becomes even more drastic. This is shown by a side-by-side comparison where each method must perform a temporal subgraph search, where the query pattern is either a three- or a four-clique, the temporal length of the query is between one and five, and the temporal behavior of the edges in the query graph are generated randomly (I.e. synthetically).

As the baseline approach is first looking for the structural matches and then refines with the temporal behavior of the edges in the query graph as opposed to the index-supported solution which uses the temporal behavior first and then expands the structural isomorphism from the central SSG on, a direct comparison between both approaches is useful: Therefore, various query graphs with three to five nodes and temporal length of four are processed using both approaches. The result is that the index-based solution (benefiting from a high selectivity in the temporal pattern of the SSG in the query) outperforms the other approach by several orders of magnitude.

For the baseline approach as well as the refinement step in the index-supported approach we rely on Ullmann's algorithm to retrieve all subgraph candidates. Ullmann proposed two filters (a filter based on the degree of nodes, and a filter dependent on whether a node's neighbors can be mapped), to which our paper brings another two novel filter approaches introduced in the previous section. All four filters should be evaluated regarding their effectiveness (I.e., how many candidates are they able to prune), their cost (how much time does it take to apply the filters) and – since they are freely combinable – their cost and effectiveness in combination. Experiments provided in the paper provide measurements and data to this question. A further look is also given to the order in which multiple filters have to be applied in.

As the experiments have shown, the index-based solution greatly outperforms basic approaches in applications where a dedicated portion of the query aspect lies on the temporal domain. In such cases, the “temporal-first” approach can quickly gather the small set of candidates, because the temporal query conditions are more selective.

Even the simplest SSG is suitable for a viable index, however if the specific application context allows for indexing more complex SSGs, the query processing can be sped up even more.

### 5.6.5 Discussion

The proposed index structure provides efficient support for mining temporal subgraphs from a temporal social network. However, it requires the selection of a Simple Subgraph Structure beforehand in the offline phase to build the index. While on the one hand a more specific, thus complex SSG is likely to provide better (i.e.: fewer) candidate results from the index, the probability increases that the SSG is not contained in the query graph in the online phase. In this case, the index becomes useless. It is therefore recommended to either choose a simple SSG (which experiments have shown to be a viable approach) or create several parallel indexes for different SSGs so that the most suitable for the query may be chosen. This may require knowledge about the application-specific query patterns, thus a quantitative evaluation has not been performed in the context of this work, which on the contrary focusses on generic applications and use cases.

## 5.7 Querying Uncertain Spatio-Temporal Data

### 5.7.1 Problem Motivation

In real-world applications observations and measurements often lack infinite preciseness. Observations may be sparse, i.e. between them lies a timeframe during which nothing is known about the measured object, or measurements themselves may be imprecise. With the publication [SIGMOD'14], a framework is introduced which aims at managing this uncertain spatio-temporal data by computing possible worlds between observations. It allows to perform uncertain temporal window queries, where the user selects a timeframe and spatial region and retrieves objects and their corresponding probability to be in this region during the specified time. Also supported are uncertain nearest-neighbor-queries, where objects can be identified that have a probability of being the nearest neighbor to a specified point in a given timespan.

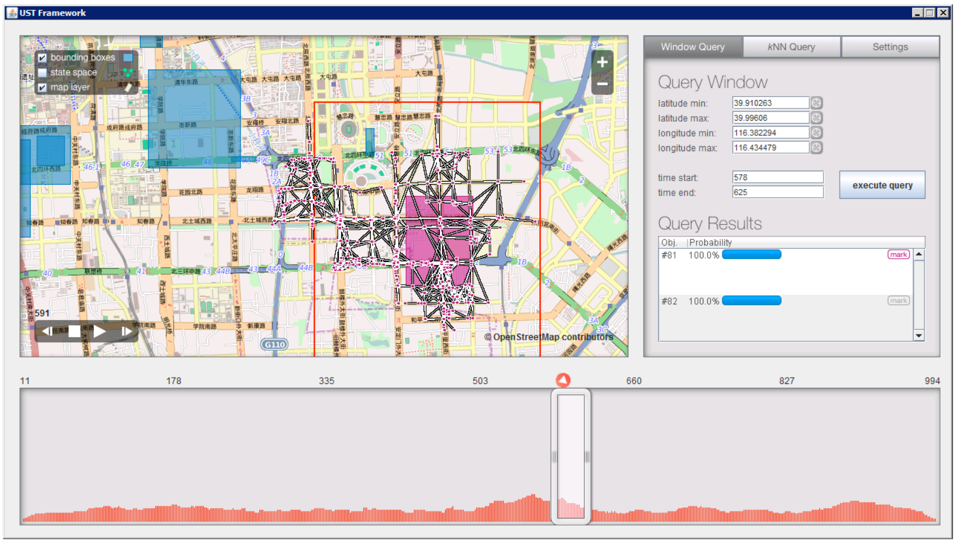
### 5.7.2 Dataset

As a dataset the *T-Drive* dataset [31] has been chosen. It consists of GPS-logfiles for over 10,000 taxicabs in Beijing for one week. For data visualization, OpenStreetMap[11] is used as an underlying map layer to better visualize trajectories, state space and spatial properties.

### 5.7.3 Framework

The framework offers a powerful visualization, as it visually combines the state space used for the Markov-models with a map layer (Figure 8). Object trajectories can be displayed and if a specified location for an object is unknown, it is represented by a bounding box which contains the object’s actual location (Figure 6 and Figure 7). By allowing to manipulate the time-axis, or using the auto-play feature, the dynamic change of these elements is graphically visualized.

[11] http://www.openstreetmap.org/



Query Window

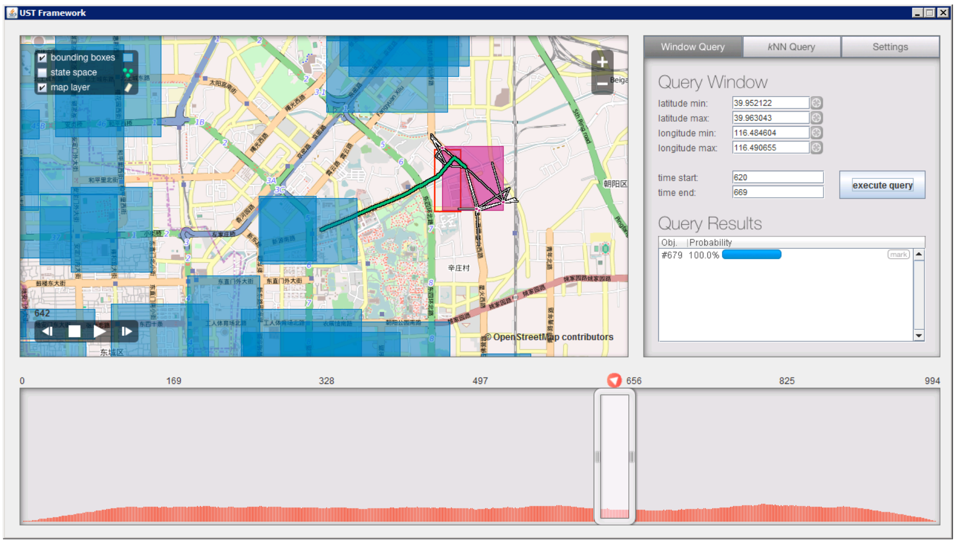
latitude min: 39.910263
latitude max: 39.99606
longitude min: 116.382294
longitude max: 116.434479

time start: 578
time end: 625

Query Results

| Obj. | Probability |
|---|---|
| #81 | 100.0% |
| #82 | 100.0% |

*Figure 6: Displaying the lifetime trajectory of the selected (purple) object. Its current exact position is unknown but bounded by the purple box. The red box represents a spatio-temporal window query, for which probabilistic results are returned.*



Query Window

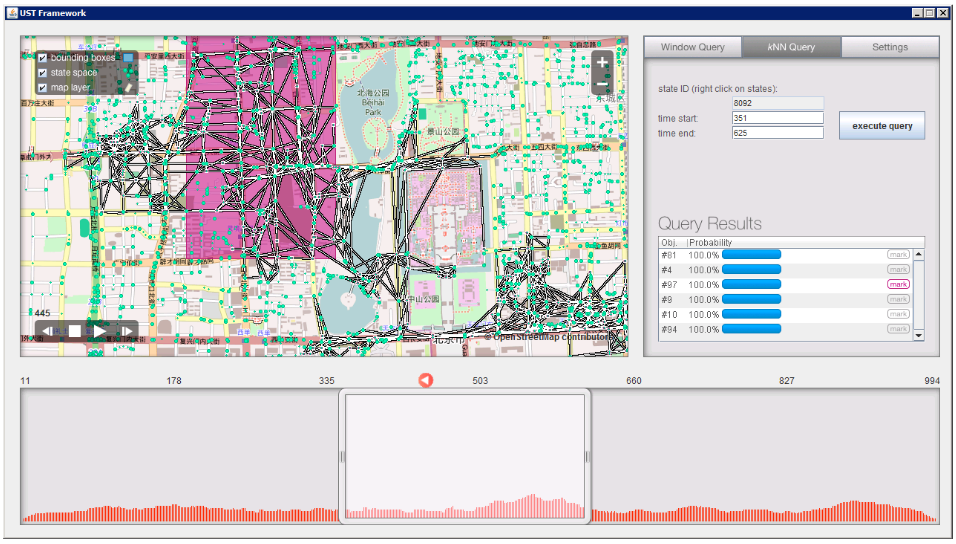
latitude min: 39.952122
latitude max: 39.963043
longitude min: 116.484604
longitude max: 116.490655

time start: 620
time end: 669

Query Results

| Obj. | Probability |
|---|---|
| #679 | 100.0% |

*Figure 7: Objects' spatial locations bounded by boxes.*



state ID (right click on states): 8092

time start: 351

time end: 625

execute query

Query Results

| Obj. | Probability | |
|---|---|---|
| #81 | 100.0% | mark |
| #4 | 100.0% | mark |
| #97 | 100.0% | mark |
| #9 | 100.0% | mark |
| #10 | 100.0% | mark |
| #94 | 100.0% | mark |

*Figure 8: State space and trajectories overlaid on actual map data.*

### 5.7.4 Discussion

The demonstration of the framework visualizes the challenges when working with uncertain spatio-temporal data and demonstrates how probabilistic queries can be handled. Analysis functionality helps to explore temporal dimension and makes uncertain or sparse datasets tangible.

## 5.8 Using Enriched Spatial Data to Compute Popular Paths

### 5.8.1 Problem Motivation

Routing mechanisms usually focus on finding the shortest, fastest or most economical route. However, when users have to remember a route, they tend to only memorize a limited number of instructions or waypoints. By giving them a small set of memorisable, significant waypoints, such as Points of Interest (POIs), navigation can be made easier. Other users may not face the problem of needing to memorize a route but may opt for a more scenic route if it just means adding little extra travel time [24]. For the later, we introduce the *Popular Path Query* in [ICDE'15], where a set of pareto-optimal routes regarding travel time and popularity is calculated. For the first use case, the *k-Constrained Popular Path Query* calculates routes along $k$ many POIs that maximize popularity. $k$ is intended to be user-definable so the user may decide how many POIs they want to remember.

### 5.8.2 Enriching the Graph with Popularity Attributes

To give a measure for the popularity of a POI, the graph is enriched with crowd-sourced information from which the popularity is deducted. A problem is encountered when dealing with popularity in a path-finding scenario: Algorithms usually try to minimize the cost of an attribute to give the efficient route. Maximizing an attribute's value may thus result in a very long, or even circular path. Thus, a method is proposed which transforms the vertex-related *gain* (popularity) into an edge-related *cost*. This allows us to adapt traditional path-finding methods to popular routes.

The *k-Constrained Popular Path Query* is then an extension of the problem: The paths are now filtered to only have results that include *at least* $k$ POIs. To measure the overall popularity of a path, only the $k$ most important POIs in a path are considered.

### 5.8.3 Framework

The developed framework supports both *Popular* Path and k-*Constrained Popular Path Queries* and gives the user a detailed visualization of the result routes, the POIs along their way and additional mined information about the POIs along the way.

### 5.8.4 Discussion

The demonstration provides a valid and novel way of computing interesting routes for users. By focussing on POIs along the way, the route becomes on the one hand more interesting and on the other hand more descriptive.

A further extension of the work may focus on how the POIs are distributed along the journey: At the local vicinity of the start and end point of the route POIs may be required in a denser sequence than along the journey: For example, a description of a route from Berlin to Paris requires more detailed description on how to reach Berlin's airport than how to take the flight from Berlin to Paris itself – i.e. the length of a track segment does not have to correspond with the breadth of its description.

# 6 Summary

This thesis has presented novel data mining and data management techniques. Various aspects of multi-representation have been examined: geospatial data spaces in combination with social data, social data combined with temporal information, uncertain spatio-temporal data, and enriched spatial data. Furthermore, a generic approach for indexing unspecified multi-metric data pursues the concept of assessing an individual data metric's properties. Data mining techniques introduced within the scope of this cumulative thesis demonstrate the valuable insights that can be mined from such heterogenous data.

# 7 References

[1] T. Emrich, M. Franzke, N. Mamoulis, M. Renz and A. Züfle, "Geo-Social Skyline Queries," in *Proceedings of the 19th International Conference on Database Systems for Advanced Applications*, Bali, Indonesia, 2014.

[2] T. Emrich, M. Franzke, H.-P. Kriegel, J. Niedermayer, M. Renz and A. Züfle, "An Extendable Framework for Managing Uncertain Spatio-Temporal Data," in *Proceedings of the 2014 ACM International Conference on Management of Data*, Snowbird, UT, United States, 2014.

[3] G. Jossé, M. Franzke, G. Skoumas, A. Züfle, M. A. Nascimento and M. Renz, "A Framework for Computation of Popular Paths from Crowdsourced Data," in *Proceedings of the 31st IEEE International Conference on Data Engineering*, Seoul, South Korea, 2015.

[4] M. Franzke, T. Emrich, A. Züfle and M. Renz, "Indexing Multi-Metric Data," in *Proceedings of the 32nd IEEE International Conference on Data Engineering*, Helsinki, Finland, 2016.

[5] M. Franzke, J. Bleicher and A. Züfle, "Finding Influencers in Temporal Social Networks Using Intervention Analysis," in *Proceedings of the 27th Australasian Database Conference*, Sydney, Australia, 2016.

[6] M. Franzke, T. Emrich, A. Züfle and M. Renz, "Pattern Search in Temporal Social Networks," in *Proceedings of the 21st International Conference on Extending Database Technology*, Vienna, Austria, 2018.

[7] S. Borzsony, D. Kossmann and K. Stocker, "The Skyline operator," in *Proceedings of the 17th International Conference on Data Engineering*, Heidelberg, Germany, 2001.

[8] X. Lin, Y. Yuan, Q. Zhang and Y. Zhang, "Selecting Stars: The k Most Representative Skyline Operator," in *Proceedings of the 23rd IEEE International Conference on Data Engineering*, Istanbul, Turkey, 2007.

[9] J. Pei, W. Jin, M. Ester and Y. Tao, "Catching the Best Views of Skyline: A Semantic Approach Based on Decisive Subspaces," in *Proceedings of the 31st International Conference on Very Large Data Bases*, Trondheim, Norway, 2005.

[10] T. Emrich, H.-P. Kriegel, N. Mamoulis, M. Renz und A. Züfle, „Querying Uncertain Spatio-Temporal Data," in *Proceedings of the 28th IEEE International Conference on Data Engineering*, Washington, DC, United States, 2012.

[11] T. Emrich, H.-P. Kriegel, N. Mamoulis, M. Renz und A. Züfle, „Indexing Uncertain Spatio-Temporal Data," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, Maui, HI, United States, 2012.

[12] J. Niedermayer, A. Züfle, T. Emrich, M. Renz, N. Mamoulis, L. Chen und H.-P. Kriegel, „Similarity Search on Uncertain Spatio-Temporal Data,“ in *Proceedings of the 6th International Conference on Similarity Search and Applications*, A Coruña, Spain, 2013.

[13] J. Niedermayer, A. Züfle, T. Emrich, M. Renz, N. Mamoulis, L. Chen and H.-P. Kriegel, "Probabilistic Nearest Neighbor Queries on Uncertain Moving Object Trajectories," *Proceedings of the VLDB Endowment,* vol. 7, no. 3, pp. 205-216, November 2013.

[14] E. Cho, S. A. Myers and J. Leskovec, "Friendship and Mobility: User Movement In Location-Based Social Networks," in *Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, United States, 2011.

[15] H. Tong, C. Faloutsos and J.-Y. Pan, "Fast Random Walk with Restart and Its Applications," in *Proceedings of the 6th IEEE International Conference on Data Mining*, Hong Kong, China, 2006.

[16] I. Konstas, V. Stathopoulos and J. M. Jose, "On Social Networks and Collaborative Recommendation," in *Proceedings of the 32nd Annual International ACM Conference on Research and Development in Information Retrieval*, Boston, MA, United States, 2009.

[17] P. Berkhin, "Bookmark-Coloring Algorithm for Personalized PageRank Computing," *Internet Mathematics,* vol. 3, no. 1, pp. 41-62, 2006.

[18] Y. Fujiwara, M. Nakatsuji, M. Onizuka and M. Kitsuregawa, "Fast and Exact Top-k Search for Random Walk with Restart," *Proceedings of the VLDB Endowment,* vol. 5, no. 5, pp. 442-453, January 2012.

[19] N. Armenatzoglou, S. Papadopoulos and D. Papadias, "A General Framework for Geo-Social Query Processing," *Proceedings of the VLDB Endowment,* vol. 6, no. 10, pp. 913-924, August 2013.

[20] P. Ciaccia und M. Patella, „The M²-tree: Processing Complex Multi-Feature Queries with Just One Index,“ in *Proceedings of the First DELOS Network of Excellence Workshop on Information Seeking, Searching and Querying in Digital Libraries*, Zurich, Switzerland, 2000.

[21] B. Bustos and T. Skopal, "Dynamic similarity search in multi-metric spaces," in *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, Santa Barbara, CA, United States, 2006.

[22] P. Ciaccia, M. Patella and P. Zezula, "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," in *Proceedings of the 23rd International Conference on Very Large Databases*, Athens, Greece, 1997.

[23] R. Fagin, „Combining Fuzzy Information from Multiple Systems," in *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, Montreal, QC, Canada, 1996.

[24] T. Bernecker, T. Emrich, F. Graf, H.-P. Kriegel, P. Kröger, M. Renz, E. Schubert und A. Zimek, „Subspace Similarity Search Using the Ideas of Ranking and Top-k Retrieval," in *Proceedings of the 26th IEEE International Conference on Data Engineering Workshops*, Long Beach, CA, United States, 2010.

[25] N. Beckmann, H.-P. Kriegel, R. Schneider and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," in *Proceedings of the 1990 ACM International Conference on Management of Data*, Atlantic City, NJ, United States, 1990.

[26] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet Physics Doklady,* vol. 10, no. 8, pp. 707-710, February 1966.

[27] G. E. P. Box and G. C. Tiao, "Intervention Analysis with Applications to Economic and Environmental Problems," *Journal of the American Statistical Association,* vol. 70, no. 349, pp. 70-79, March 1975.

[28] S. R. Khandker, G. B. Koolwal and H. A. Samad, Handbook on Impact Evaluation: Quantitative Methods and Practices, Washington, DC, United States: The World Bank, 2010.

[29] N. S. Fleming, E. Gibson and D. G. Fleming, "The Use of Proc ARIMA to Test an Intervention Effect," in *SAS Conference Proceedings: South-Central SAS Users Group*, Houston, TX, United States, 1997.

[30] F. Sowell, "Maximum likelihood estimation of stationary univariate fractionally integrated time series models," *Journal of Econometrics,* vol. 53, no. 1-3, pp. 165-188, July-September 1992.

[31] J. R. Ullmann, "An Algorithm for Subgraph Isomorphism," *Journal of the ACM,* vol. 23, no. 1, pp. 31-42, 1976.

[32] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun and Y. Huang, "T-Drive: Driving Directions Based on Taxi Trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, San Jose, CA, United States, 2010.

[33] D. Quercia, R. Schifanella and L. M. Aiello, "The Shortest Path to Happiness: Recommending Beautiful, Quiet, and Happy Routes in the City," in *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, Santiago, Chile, 2014.