Urban Computing: Concepts, Methodologies, and Applications

YU ZHENG, Microsoft Research
LICIA CAPRA, University College London
OURI WOLFSON, University of Illinois at Chicago
HAI YANG, Hong Kong University of Science and Technology

Urbanization's rapid progress has modernized many people's lives but also engendered big issues, such as traffic congestion, energy consumption, and pollution. Urban computing aims to tackle these issues by using the data that has been generated in cities (e.g., traffic flow, human mobility, and geographical data). Urban computing connects urban sensing, data management, data analytics, and service providing into a recurrent process for an unobtrusive and continuous improvement of people's lives, city operation systems, and the environment. Urban computing is an interdisciplinary field where computer sciences meet conventional city-related fields, like transportation, civil engineering, environment, economy, ecology, and sociology in the context of urban spaces. This article first introduces the concept of urban computing, discussing its general framework and key challenges from the perspective of computer sciences. Second, we classify the applications of urban computing into seven categories, consisting of urban planning, transportation, the environment, energy, social, economy, and public safety and security, presenting representative scenarios in each category. Third, we summarize the typical technologies that are needed in urban computing into four folds, which are about urban sensing, urban data management, knowledge fusion across heterogeneous data, and urban data visualization. Finally, we give an outlook on the future of urban computing, suggesting a few research topics that are somehow missing in the community.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data mining, Spatial databases and GIS*; J.2 [**Physical Sciences and Engineering**]: *Earth and atmospheric sciences, Mathematics and statistics*; J.4 [**Social and Behavioral Sciences**]: *Economics, Sociology*; G.1.6 [**Optimization**]; G.1.2 [**Approximation**]; E.1 [**Data Structures**]; E.2 [**Data Storage Representations**]

General Terms: Algorithms, Measurement, Experimentation

Additional Key Words and Phrases: Urban computing, urban informatics, big data, human mobility, city dynamics, urban sensing, knowledge fusion, computing with heterogeneous data, trajectories

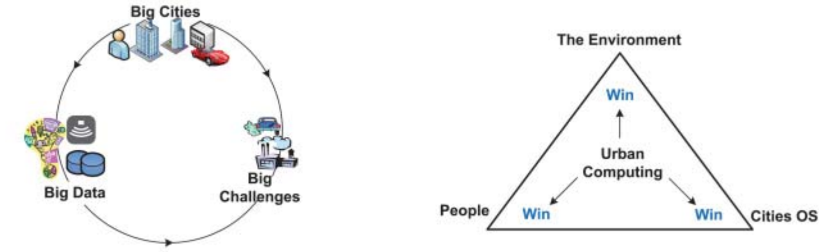
ACM Reference Format:
Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. ACM Trans. Intell. Syst. Technol. 5, 3, Article 38 (September 2014), 55 pages.
DOI: http://dx.doi.org/10.1145/2629592

1. INTRODUCTION

Urbanization's rapid progress has led to many *big cities*, which have modernized many people's lives but also engendered *big challenges*, such as air pollution, increased energy consumption, and traffic congestion. Tackling these challenges seemed nearly

Authors' addresses: Y. Zheng, Microsoft Research, Building 2, No. 5 Danling Street, Haidian District, Beijing 100080, China; email: yuzheng@microsoft.com; L. Capra, Dept. of Computer Science, University College London, Gower Street, London WC1E 6BT, United Kingdom; email: ucaclca@ucl.ac.uk; O. Wolfson, Department of Computer Sciences, University of Illinois at Chicago, Chicago, Illinois; email: wolfson@cs.uic.edu; H. Yang, Department of the Civil and Environmental Engineering, Hong Kong University of Sciences and Technology, Clear Water Bay, Kowloon, Hong Kong; email: cehyang@ust.hk.

DOI: http://dx.doi.org/10.1145/2629592



(a) Motivation: Big cities, data and challenges (b) Goal of urban computing

Fig. 1. Motivation and goal of urban computing.

impossible years ago given the complex and dynamic settings of cities. Nowadays, sensing technologies and large-scale computing infrastructures have produced a variety of *big data* in urban spaces (e.g., human mobility, air quality, traffic patterns, and geographical data). The *big data* implies rich knowledge about a city and can help tackle these challenges when used correctly. For instance, we can detect the underlying problems in a city's road network through analyzing the city-wide human mobility data. This discovery can help better formulate city planning for the future [Zheng et al. 2011b]. Another example is to exploit the root cause of urban air pollution by studying the correlation between air quality and other data sources, such as traffic flow and points of interest (POIs) [Zheng et al. 2013b].

Motivated by the opportunities of building more intelligent cities, we came up with a vision of urban computing, which aims to unlock the power of knowledge from big and heterogeneous data collected in urban spaces and apply this powerful information to solve major issues our cities face today [Zheng et al. 2012c, 2013a]. In short, we are able to tackle the *big challenges* in *big cities* by using *big data*, as depicted in Figure 1(a).

Though the term urban computing has been used before [Kindberg et al. 2007; Kostakos and O'Neill 2008], it is still a vague concept with many questions open. For example, what are the core research problems of urban computing? What are the challenges of the research theme? What are the key methodologies for urban computing? What are the representative applications in this domain, and how does an urban computing system work?

To address these issues, we formally coin urban computing in this article and introduce its general framework, key research problems, methodologies, and applications. This article will help the community better understand and explore this nascent area, therefore generating quality research results and real systems that can eventually lead to greener and smarter cities. In addition, urban computing is a multidisciplinary research field, where computer sciences meet conventional city-related areas, such as civil engineering, transportation, economics, energy engineering, environmental sciences, ecology, and sociology. This article mainly discusses the aforementioned problems from the perspective of computer sciences.

The rest of the article is organized as follows. In Section 2, we introduce the concept of urban computing, presenting a general framework and the key challenges of each step in the framework. The datasets that are frequently used in urban computing are also discussed. In Section 3, we categorize the applications of urban computing into seven groups, presenting some representative scenarios in each group. In Section 4, we introduce four folds of methodologies that are usually employed in an urban computing scenario. In Section 5, we conclude the article and point out a few future directions of this research theme.

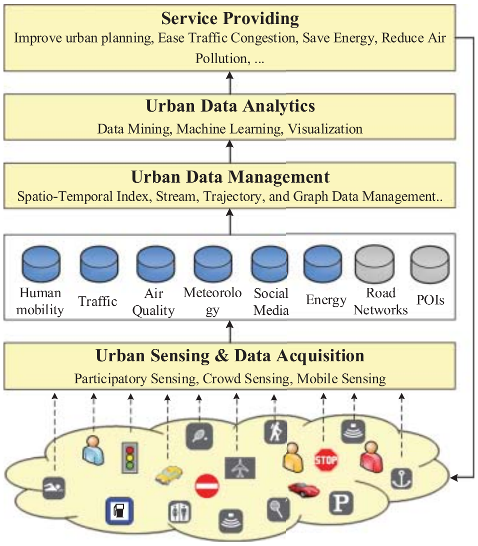

Service Providing
Improve urban planning, Ease Traffic Congestion, Save Energy, Reduce Air Pollution, ...
Urban Data Analytics
Data Mining, Machine Learning, Visualization
Urban Data Management
Spatio-Temporal Index, Stream, Trajectory, and Graph Data Management..
Human mobility
Traffic
Air Quality
Meteorology
Social Media
Energy
Road Networks
POIs
Urban Sensing & Data Acquisition
Participatory Sensing, Crowd Sensing, Mobile Sensing
STOP
P

Fig. 2. General framework of urban computing.

2. FRAMEWORK OF URBAN COMPUTING

2.1. Definition

Urban computing is a process of acquisition, integration, and analysis of big and heterogeneous data generated by diverse sources in urban spaces, such as sensors, devices, vehicles, buildings, and humans, to tackle the major issues that cities face (e.g., air pollution, increased energy consumption, and traffic congestion). Urban computing connects unobtrusive and ubiquitous sensing technologies, advanced data management and analytic models, and novel visualization methods to create win-win-win solutions that improve urban *environment*, *human* life quality, and *city* operation systems, as shown in Figure 1(b). Urban computing also helps us understand the nature of urban phenomena and even predict the future of cities. Urban computing is an interdisciplinary field fusing the computing science field with traditional fields like transportation, civil engineering, economy, ecology, and sociology in the context of urban spaces.

2.2. General Framework

Figure 2 depicts a general framework of urban computing, which is composed of four layers: urban sensing, urban data management, data analytics, and service providing. Using urban anomaly detection as an example [Pan et al. 2013], we briefly introduce the operation of the framework as follows.

In the urban sensing step, we constantly probe people's mobility (e.g., routing behavior in a city's road network) using GPS sensors or their mobile phone signals. We also continuously collect the social media people have posted on the Internet. In the data management step, the human mobility and social media data are well organized by some indexing structure that simultaneously incorporates spatiotemporal information and texts for supporting efficient data analytics. In the data analytics step, once an anomaly occurs, we are able to identify the locations where people's mobility significantly differs from its origin patterns. In the meantime, we can describe the anomaly by mining representative terms from the social media that is related to the locations and time span. In the service providing step, the locations and description of the anomaly will be sent to the drivers nearby so that they can choose a bypass. In addition, the information will be delivered to the transportation authority for dispersing traffic and diagnosing the anomaly. The system continues the loop for an instant and unobtrusive detection of urban anomalies, helping improve people's driving experiences and reduce traffic congestion.

Compared with other systems (e.g., web search engines) that are based on a single (modal)-data/single-task framework (i.e., information retrieval from web pages), urban computing holds a multi(modal)-data/multitask framework. The tasks of urban computing include improving urban planning, easing traffic congestion, reducing energy consumption, and reducing air pollution. Additionally, we usually need to harness a diversity of data sources in a single task. For instance, the aforementioned anomaly detection uses human mobility data, road networks, and social media. Different tasks can be fulfilled by combining different data sources with different data acquisition, management, and analytics techniques from different layers of the framework.

2.3. Key Challenges

The goals and framework of urban computing result in three main challenges:

1. ***Urban sensing and data acquisition*:** The first is data acquisition techniques that can *unobtrusively* and *continually* collect data in a *citywide* scale. This is a nontrivial problem given the three italic terms. Monitoring the traffic flow on a road segment is easy, but *continually* probing the *citywide* traffic is challenging as we do not have sensors on every road segment. Building new sensing infrastructures could achieve the goal but in turn would aggravate the burden of cities. How to leverage what we already have in urban spaces intelligently has yet to be explored. Humans as a sensor is a new concept that may help tackle this challenge. For instance, when users post social media on a social networking site, they are actually helping us understand the events happening around them. When many people drive on a road network, their GPS traces may reflect the traffic patterns and anomalies. However, just as a coin has two sides, despite the flexibility and intelligence of human sensors, humans as a sensor also brings three challenges (we will discuss more about this in Section 4.1):
   - *Energy consumption and privacy*: This is a nontrivial problem for participatory sensing applications, where users proactively contribute their data (usually using a smartphone) to save the energy of a smartphone and protect their privacy during the sensing process. There is a tradeoff among energy, privacy, and the utility of shared data [Xue et al. 2013].
   - *Loose-controlled and nonuniform distributed sensors*: We can put traditional sensors anywhere we like and configure these sensors to send sensing readings at a certain frequency. However, we cannot control people who would send information any time they like or do not share data sometimes. In some places, we may not even have people at some moments (i.e., may not have sensor data), inevitably resulting in data-missing and sparsity problems. On the other hand, the user-generated content in some location (with many people) may be oversufficient or even redundant, adding unnecessary workload for sensing, communication, and storage. Additionally, what we can obtain is always a sample of data from partial users, as not everyone shares data. The distribution of the sample data may be skewed from the distribution of the entire dataset, depending on the movement of people.
   - *Unstructured, implicit, and noise data*: The data generated by traditional sensors is well structured, explicit, clean, and easy to understand. However, the data contributed by users is usually in a free format, such as texts and images, or

cannot explicitly lead us to the final goal as when using traditional sensors. Sometimes, the information from human sensors is also quite noisy.

Using the application presented in Zhang et al. [2013] as an example, we illustrate the two challenges. In this example, Zhang et al. aim to use GPS-equipped taxi drivers as sensors to detect the queuing time in a gas station (when they are refueling taxis) and further infer the number of people who are also refueling their vehicles there. The goal is to estimate the gas consumption of a station and finally the citywide gas consumption in a given time span. In this application, what we obtain is the GPS trajectories of a taxi driver, which does not tell us the result explicitly. In addition, we cannot guarantee having a taxi driver in each gas station any time, which results in a data-missing problem. In the meantime, the presence of taxis in a station may be quite different from that of other vehicles (i.e., the skewed distribution); for example, observing more taxis in a gas station does not denote more other vehicles. Furthermore, taxi drivers may park taxis somewhere close to a gas station just to have a rest or wait for a traffic light. These observations from the GPS trajectory data are noisy. In short, we usually need to learn what we really need from partial, skewed, noisy, and implicit data generated by human sensors.

2. ***Computing with heterogeneous data*:**
   - *Learn mutually reinforced knowledge from heterogeneous data*: Solving urban challenges involves a broad range of factors (e.g., exploring air pollutions involves the simultaneous study of traffic flow, meteorology, and land uses). However, existing data-mining and machine-learning techniques usually handle one kind of data; for example, computer vision is dealing with images, and natural language processing is based on texts. According to studies [Zheng et al. 2013b; Yuan et al. 2012], equally treating features extracted from different data sources (e.g., simply putting these features into a feature vector and throwing them into a classification model) does not achieve the best performance. In addition, using multiple data sources in an application leads to a high-dimension space, which usually aggravates the data sparsity problem. If not handled correctly, more data sources would even compromise the performance of a model. This calls for advanced data analytics models that can learn mutually reinforced knowledge among multiple heterogeneous data generated from different sources, including sensors, people, vehicles, and buildings. See Section 4.1 for more details.
   - *Both effective and efficient learning ability*: Many urban computing scenarios (e.g., detecting traffic anomalies and monitoring air quality) require instant answers. Besides just increasing the number of machines to speed up the computation, we need to aggregate data management and mining and machine-learning algorithms into a computing framework to provide both an effective and efficient knowledge discovery ability. In addition, traditional data management techniques are usually designed for a single modal data source. An advanced management methodology that can organize multimodal data (such as streaming, geospatial, and textual data) well is still missing. So, computing with multiple heterogeneous data is a fusion of data and algorithms. See Section 4.3 for more discussion.
   - *Visualization*: Massive data brings a tremendous amount of information that needs a better presentation. A good visualization of original data could inspire new ideas to solve a problem, while the visualization of computing results can reveal knowledge intuitively so as to help in decision making. The visualization of data may also suggest the correlation or causality between different factors. The multimodal data in urban computing scenarios leads to high dimensions of views, such as spatial, temporal, and social, for a visualization. How to interrelate different kinds of data in different views and detect patterns and trends is

challenging. In addition, when facing multiple types and huge volumes of data, seeing how exploratory visualization [Andrienko et al. 2003] can provide an interactive way for people to generate new hypotheses becomes even more difficult. This calls for an integration of instant data-mining techniques into a visualization framework, which is still missing in urban computing.

3. ***Hybrid systems blending the physical and virtual worlds*:** Unlike a search engine or a digital game where the data was generated and consumed in the digital world, urban computing usually integrates the data from both worlds (e.g., combining traffic with social media). Alternatively, the data (e.g., GPS trajectories of vehicles) is generated in the physical world and then sent back to the digital world, such as a cloud system. After the data is processed with other data sources in the cloud, the knowledge learned from the data will be used to serve users from the physical world via mobile clients (e.g., driving direction suggestions, taxi ridesharing, and air quality monitoring). The design of such a system is much more challenging than for conventional systems that only reside in one world, as the system needs to communicate with many devices and users simultaneously and send and receive data of different formats and at different frequencies.

2.4. Urban Data

In this section, we introduce the frequently used data sources in urban computing and briefly mention the issues we usually face when using these data sources.

2.4.1. Geographical Data. Road network data may be the most frequently used geographical data in urban computing scenarios, for example, traffic monitoring and prediction [Pan and Zheng et al. 2013], urban planning [Zheng et al. 2011b], routing [Yuan and Zheng et al. 2010a, 2011b, 2013b], and energy consumption analysis [Zhang et al. 2013]. It is usually represented by a graph that is composed of a set of edges (denoting road segments) and a collection of nodes (standing for road intersections). Each node has unique geospatial coordinates; each edge is described by two nodes (sometimes also called terminals) and a sequence of intermediate geospatial points. Other properties, such as the length, speed constraint, type of road, and number of lanes, are associated with an edge.

A POI, such as a restaurant or a shopping mall, is usually described by a name, address, category, and set of geospatial coordinates. While there are massive POIs in a city, the information of POIs could vary in time (e.g., a restaurant may change its name, be moved to a new location, or even be shut down). As a result, collecting POI data is not an easy task. Generally, there are two approaches to produce POI data. One is obtained through existing Yellow Page data. The geospatial coordinates of an entity are automatically translated from its text address by using a geo-coding algorithm. The other approach is to manually collect POI information in the real world, for instance, carrying a GPS logger to record the geospatial coordinates of a POI. The latter approach is mainly done by some map data providers, such as Navinfo and AutoNavi. Recently, some location-based social networking services, like Foursquare, have allowed end-users to create a new POI in the system if the POI has not been included. In order to have a large coverage of POIs, the widely used online map services, like Bing and Google maps, usually combine the aforementioned two approaches to collect POI data. As a result, quite a few issues have been generated. For example, how can we verify whether the information of a POI is correct? Sometimes, the geospatial coordinates of a POI may be inaccurate, leading people to a wrong place. Or, how can we merge the POI data generated from different sources or approaches [Zheng 2010c]?

Land use data describes the function of a region, such as residential areas, suburban areas, and forests, originally planned by urban planners and roughly measured by satellite images in practice. For example, the U.S. Geological Survey categorizes each 30 m × 30 m square of the United States into 21 types of ground cover [USA Ground Cover], such as grass land, water, and commercial. In many developing countries where cities change over time with many new infrastructures built and old buildings removed, the reality of a city may be different from its original planning. As the satellite image cannot differentiate between fine-grained land use categories, such as educational, commercial, and residential areas, obtaining the current land use data of a big city is not easy [Yuan and Zheng et al. 2012a].

2.4.2. Traffic Data. There are many ways to collect traffic data, such as using loop sensors, surveillance cameras, and floating cars. Loop sensors are usually embedded in pairs in major roads (e.g., highways). Instead of recording the absolute time, such sensors detect the time interval that a vehicle travels across two consecutive (i.e., a pair of) detectors. Knowing the distance between a pair of loop detectors, we can calculate the travel speed on the road based on the time interval. Counting the number of vehicles traversing a pair of loop detectors in a time slot, we know the traffic volume on a road. As deploying and maintaining loop sensors is very expensive in terms of money and human resources, such traffic monitoring technology is usually employed for major roads rather than low-level streets. As a result, the coverage of loop sensors is quite limited. Additionally, the loop sensor data does not tell us how a vehicle travels on a road and between two roads. Consequently, the travel time that a vehicle spends at an intersection (e.g., waiting for traffic lights and direction turns) cannot be recognized from this kind of sensor data.

Surveillance cameras are widely deployed in urban areas, generating a huge volume of images and videos reflecting traffic patterns. The data provides a visual ground truth of traffic conditions to people. However, it is still a challenging task to automatically turn the images and videos into a specific traffic volume and travel speed. It is difficult to apply a machine-learning model trained for one location to other locations, because of the differing structure of roads and differing camera settings, such as height (to the ground), angle, and focus. As a result, monitoring citywide traffic conditions through this approach is mainly based on human effort.

Floating car data is generated by vehicles traveling around a city with a GPS sensor. The trajectories of these vehicles will be sent to a central system and matched to a road network for deriving speeds on road segments. As many cities have already installed GPS sensors in taxicabs, buses, and logistics trucks for different purposes, floating car data has already been widely available. In contrast to loop sensors and surveillance camera-based approaches, floating car-based traffic monitoring methods have higher flexibility and a lower deployment cost. However, the coverage of floating car data depends on the distribution of the probing vehicles, which may change over time and be skewed in a city in a time span. In other words, the data sparsity problem still exists, calling for advanced knowledge discovery technology that can recover the citywide traffic conditions based on limited data. Castro et al. [2013] presents a survey on turning GPS trajectories of taxis into social and community dynamics.

2.4.3. Mobile Phone Signals. A call detail record (CDR) is a data record produced by a telephone exchange containing attributes that are specific to a single instance of a phone call, such as the phone numbers of both the calling and receiving parties, the start time, and the duration of that call. Having such kind of data, we can study the behavior of an individual or build a network between different users. The similarity between users can also be inferred. Another category of mobile phone signals is more concerned about the location of a user rather than the communication between phones. Using a triangle positioning algorithm, a mobile phone's location can be roughly calculated based on three or more base stations. This kind of data denotes citywide human mobility, which can be used for detecting urban anomalies or, in the long run, for studying a city's functional regions and urban planning. Sometimes the two kinds of mobile phone data are integrated (i.e., have transaction records between phones and the location of each phone).

2.4.4. Commuting Data. People traveling in cities generate a huge volume of commuting data, such as the card swiping data in a subway system or bus line and the ticketing data in parking lots. Card swiping data is widely available in a city's public transportation systems, where people swipe an RFID card when entering into a subway station or getting on a bus. Some systems also require people to swipe their cards again when leaving a station or getting off a bus. Each transaction record consists of a timestamp of entering/leaving a station and the ID of the station as well as the fare for this trip. This is another kind of data representing citywide human mobility.

Street-side parking is usually paid for through a parking meter. The payment information of parking slots may include the time the ticket is issued and the parking fare. The data indicates the traffic of vehicles around a place, which can be used to not only improve a city's parking infrastructure but also analyze people's travel patterns. The latter can support geo-ads and location choosing for a business.

2.4.5. Environmental Monitoring Data. Meteorological data includes humidity, temperature, barometer pressure, wind speed, and weather conditions, which can be crawled from public websites. Air quality data, such as the concentration of $PM_{2.5}$, NO_2, and SO_2, can be obtained from air quality monitoring stations. Some gasses like CO_2 and CO can even be detected by portable sensors. When communicating with people, air quality is represented by an Air Quality Index (AQI) and a category, for example, good, moderate, and unhealthy. Influenced by multiple complex factors, such as traffic flow and land uses, urban air quality varies significantly by location and changes tremendously over time. As a consequence, a limited number of monitoring stations cannot reveal the fine-grained air quality throughout a city.

Noise data is another kind of environmental data that has a direct impact on people's mental and physical health. Measuring noise pollution depends on both the intensity of noises and people's tolerance to noises [Zheng et al. 2014a]; the latter changes over time. In New York City, there is a 311 platform where people can complain about something imperfect (but not urgent) by making a phone call. Each complaint is associated with a timestamp, a location, and a category. Noise is the third largest category in the data. The data can be used to diagnose a city's noise pollution.

Satellite remote sensing scans the surface of the earth with rays of different lengths to generate images representing the ecology and meteorology of a wide region.

2.4.6. Social Network Data. Social network data consists of two parts. One is a social structure, represented by a graph, denoting the relationship, interdependency, or interaction between users. The other is user-generated social media, such as texts, photos, and videos, which contain rich information about a user's behaviors/interests. When adding a location to social media [Zheng et al. 2011a] (e.g., check-in data from Foursquare and geo-tagged tweets), we can model people's mobility in urban areas, which helps us detect and understand urban anomalies [Lee et al. 2010; Pan and Zheng et al. 2013].

2.4.7. Economy. There is a variety of data representing a city's economic dynamics (e.g., transaction records of credit cards, stock prices, housing prices, and people's incomes). When used aggregately, these datasets can capture the economic rhythm of a city, therefore predicting future economy.

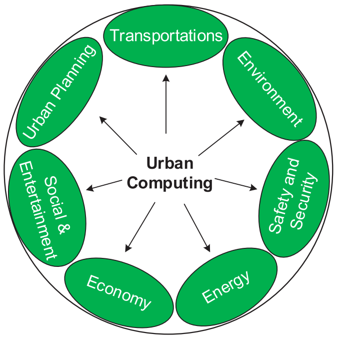


Fig. 3. Main categories of applications in urban computing.

2.4.8. Energy. The gas consumption of vehicles on road surfaces and in gas stations reflects a city's energy consumption. The data can be obtained directly from sensors (e.g., some insurance companies have been collecting different kinds of sensor data from a vehicle) or inferred from other data sources implicitly (e.g., from the GPS trajectory of a vehicle). The data can be used to evaluate a city's energy infrastructures (e.g., the distribution of gas stations), calculate the pollution emission from vehicles on road surfaces, or find the most gas-efficient driving route. Additionally, the electricity consumption of an apartment or a building can be used to optimize residential energy usage, shifting peak loads to periods of low demand.

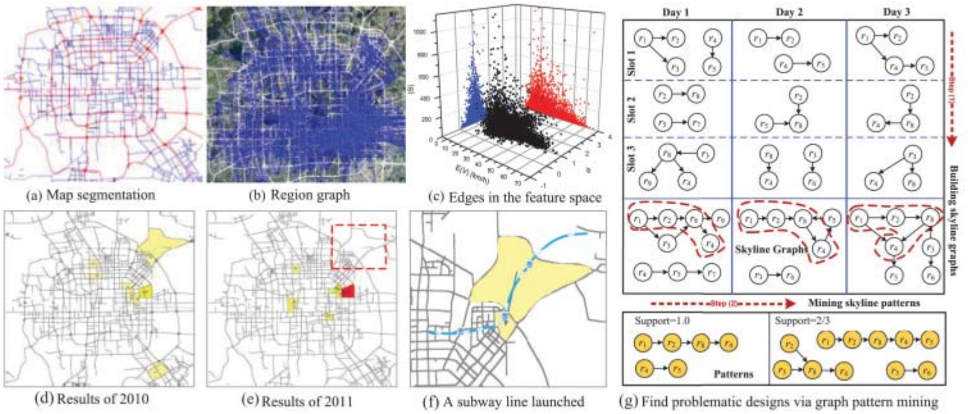
2.4.9. Health Care. There are already abundant health care and disease data generated by hospitals and clinics. In addition, the advances of wearable computing devices enable people to monitor their own health conditions, such as heart rate, pulse, and sleep time. The data can even be sent to a cloud for diagnosing a disease and doing a remote medical examination. Aside from studying an individual's health conditions, in urban computing, we can use these datasets aggregately to study the impact of environmental change on people's health. For example, how is air pollution related to the asthma situation in Hong Kong? How can urban noise impact people's mental health in New York City?

3. APPLICATIONS IN URBAN COMPUTING

Before presenting the frequently used technology in urban computing, we first list seven categories of urban computing scenarios for urban planning, transportation, environment, energy, social, economy, and public safety and security, as illustrated in Figure 3. We select some representative applications in each category, mainly focusing on its goal, motivation, results, and the data used. The methodology of each application is briefly mentioned here but will be discussed more in Section 5.

3.1. Urban Computing for Urban Planning

Effective planning is of great importance to building an intelligent city. Formulating urban planning requires evaluating a broad range of factors, such as traffic flow, human mobility, points of interest, and road network structures. These complex and fast-evolving factors turn urban planning into a very challenging task. Traditionally, urban planners relied on labor-intensive surveys to inform their decision making. For example, to understand urban commuting patterns, a series of research studies have been done based on travel survey data [Hanson and Hanson 1980; Gandia 2012; Jiang et al. 2012]. The information obtained through the surveys may not be sufficient and timely enough. Recently, the widely available human mobility data generated in



(a) Map segmentation (b) Region graph (c) Edges in the feature space (d) Results of 2010 (e) Results of 2011 (f) A subway line launched (g) Find problematic designs via graph pattern mining

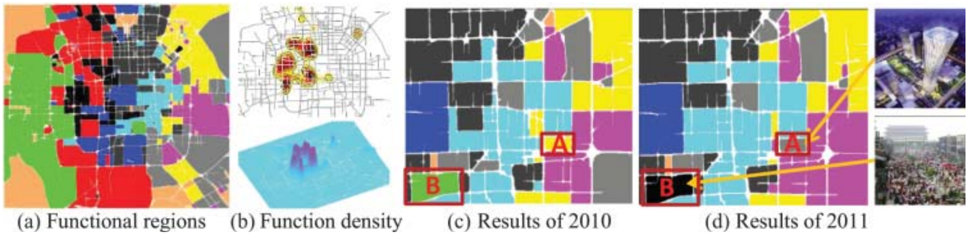
Fig. 4. Finding the underlying problem of Beijing's road network using taxi trajectories.

urban spaces has actually reflected the underlying problems of a city, providing urban planners with opportunities to better formulate future planning.

3.1.1. Gleaning Underlying Problems in Transportation Networks. Zheng et al. [2011b] gleaned the underlying problems in Beijing's transportation network by analyzing the GPS trajectories generated by 33,000 taxicabs over a period of 3 years.

They first partitioned the urban areas of Beijing into disjoint regions using major roads, such as highways and arterial roads [Yuan et al. 2012b], as illustrated in Figure 4(a). The pick-up and drop-off points of passengers were extracted from each taxi trajectory to formulate the origin–destination (OD) transitions between these regions. A region graph was then built based on the OD transitions, where a node was a region and an edge represented the aggregation of the transitions between two regions, as depicted in Figure 4(b). Using a data-driven method, a day was divided into a few time spans, which corresponded to morning rush hours, evening peak hours, and the rest. For each time span, a region graph was built based on the taxi trajectories falling into the time span. As demonstrated in Figure 4(c), three features, consisting of the volume of taxis ($|S|$), average speed of these taxis $E(V)$, and a detour ratio θ, were extracted for each edge based on the associated taxi trajectories. Representing an edge with a point in the three feature dimension spaces, the points with large $|S|$, small $E(V)$, and big θ could be underlying problems. That is, the connection between two regions was not effective enough to support the traffic traveling between them, resulting in a large volume, low speed, and big detour ratio.

Using a skyline algorithm, a set of points (called skyline edges) can be detected from the data of each time slot. As illustrated in Figure 4(g), the skyline edges from different time slots of the same day were connected to formulate skyline graphs if they were spatially overlapped by some nodes and temporally adjacent. Finally, some subgraph patterns can be obtained through mining the skyline graph across multiple days; for example, $r_1 \rightarrow r_2 \rightarrow r_8 \rightarrow r_4$ occurred on all 3 days. Such graph patterns represent the underlying problems in a road network, showing the correlations between individual regions and avoiding the false alerts that could be caused by some traffic accidents. By comparing the results detected from 2 consecutive years, the research can even evaluate if a newly built transportation facility works well. As demonstrated in Figures 4(d), 4(e), and 4(f), the underlying problem detected in 2010 disappeared in 2011 because of a newly launched subway line. In short, the subway line worked well in resolving the problem.



(a) Functional regions (b) Function density (c) Results of 2010 (d) Results of 2011

Fig. 5. Identifying functional regions in a city using human mobility and POIs.

3.1.2. Discover Functional Regions. The development of a city gradually fosters different functional regions, such as educational areas and business districts, which support different needs of people's urban lives and serve as a valuable organizing technique for framing detailed knowledge of a metropolis. These regions may be artificially designed by urban planners or naturally formulated according to people's actual lifestyles and would change their functions and territories with the development of a city. The understanding of functional regions in a city can calibrate urban planning and facilitate other applications, such as choosing a location for a business.

Yuan et al. [2012a] proposed a framework (titled DRoF) that Discovers Regions of different Functions in a city using human mobility between regions and POIs located in a region. For example, the red regions shown in Figure 5(a) denote the educational and scientific areas of Beijing. However, the function of a region is compound rather than single, represented by a distribution across multiple functions. The regions with the same color actually share a similar distribution of functions. On the other hand, even if a region is recognized as an educational area, it does not mean every part of the region serves this function. For instance, there could be some shopping centers around a university. So, given a function, Yuan et al. further identified its kernel density distribution [Wand and Jones 1995]. Figure 5(b) shows the density distribution of commercial areas in Beijing; the darker the area is, the higher the probability is that the location could be a commercial area. In their methodology, a city was segmented into disjointed regions according to major roads, such as highways and urban expressways. They infer the functions of each region using a topic-based inference model, which regards a region as a document, a function as a topic, categories of POIs (e.g., restaurants and shopping malls) as metadata (like authors, affiliations, and keywords), and human mobility patterns (when people reach/leave a region and where people come from and leave for) as words. As a result, a region is represented by a distribution of functions, each of which is further denoted by a distribution of mobility patterns. Here, human mobility can differentiate between the popularities of POIs belonging to the same category. It also indicates the function of a region; for example, people leave residential areas in the morning and return in the evening. Specifically, the human mobility data was extracted from the GPS trajectories generated by over 33,000 taxis over a period of 3 months in 2010 and 2012, respectively. Finally, nine kinds of functional regions were identified based on the clustering results and human labeling.

There are other approaches to this problem. For example, Toole et al. [2012] utilized call detail records, which provide information on the location of mobile phones any time a call is made or a text message is sent, to measure spatiotemporal changes in phone activities. Using a classification algorithm, they inferred the land use of a region based on the dynamic phone activity patterns in the region. Three weeks of call records for roughly 600,000 users in the Boston region were used to infer four kinds of land uses. Different from Yuan's method, which is an unsupervised learning algorithm, Toole et al. approached the problem with a supervised learning algorithm. In another example, using a database approach, Sheng et al. [2010] searched for some regions with a similar distribution of POIs to a given region. Since POI data is very important in determining the function of a region, ensuring its quality (e.g., matching and merging POIs from different sources) is also a practical problem [Zheng et al. 2010c].

3.1.3. Detecting a City's Boundary. The regional boundaries defined by governments may not respect the natural ways that people interact across space. The discovery of the *real* borders of regions according to the interaction between people can provide decision support tools for policy makers, suggesting optimal administrative borders for a city. The discovery also helps government understand the evolving of a city's territory. The general idea of this category of research is to first build a network between locations based on human interaction (e.g., GPS tracks or phone call records) and then partition the network using some community discovery method, which finds some location clusters with denser interaction between locations in the cluster than between clusters.

Ratti et al. [2010] proposed a fine-grained approach to regional delineation through analyzing the human network inferred from a large telecommunications database in Great Britain. Given a geographical area and some measure of the strength of links between its inhabitants, they partitioned the area into smaller, nonoverlapping regions while minimizing the disruption to each person's links. The algorithm yielded geographically cohesive regions that correspond with administrative regions while unveiling unexpected spatial structures that had previously only been hypothesized in the literature.

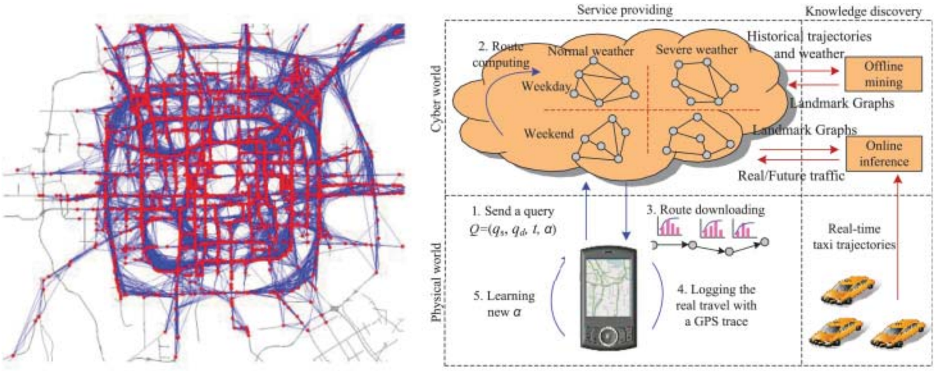
Rinzivillo et al. [2012] addressed the problem of finding the borders of human mobility at the lower spatial resolution of municipalities or counties. They mapped vehicle GPS tracks onto regions to formulate a complex network in Pisa. A community discovery algorithm, namely, Infomap, was then used to partition the network into nonoverlapped subgraphs.

3.2. Urban Computing for Transportation Systems

3.2.1. Improving Driving Experiences. Finding fast driving routes saves both the time of a driver and energy consumption as traffic congestion wastes a lot of gas [Hunter et al. 2009; Kanoulas et al. 2006]. Intensive studies have been done to learn historical traffic patterns [Bejan et al. 2010; Herrera et al. 2010], estimate real-time traffic flows [Herring et al. 2010], and forecast future traffic conditions [Castro-Neto et al. 2009] on individual road segments in terms of floating car data [Pfoser 2008, Pfoser et al. 2008], such as GPS trajectories of vehicles, WiFi, and GSM signals. However, work modeling the citywide traffic patterns is still rare.

VTrack [Thiagarajan et al. 2009] is a system for travel time estimation based on WiFi signals, measuring and localizing the time delays. The system uses a hidden Markov model (HMM)-based map matching scheme that interpolates sparse data to identify the most probable road segments driven by the user. A travel time estimation method is then proposed to attribute travel times to those segments. The experiments show that VTrack can tolerate significant noise and outages in these location estimates and still successfully identify delay-prone segments.

T-Drive [Yuan and Zheng et al. 2010a, 2011b, 2013b] is a system that provides personalized driving directions that adapt to weather, traffic conditions, and a person's own driving habits. The first version of this system [Yuan and Zheng et al. 2010a] only suggests the practically fastest path based on historical trajectories of taxicabs. The key insights consist of two parts: (1) GPS-equipped taxicabs can be regarded as mobile sensors continually probing the traffic patterns on road surfaces, and (2) taxi drivers are experienced drivers who can find a really quick route based on their knowledge, which incorporates not only the distance of a route but also the traffic conditions



(a) Landmark graph of Beijing (k=4000)

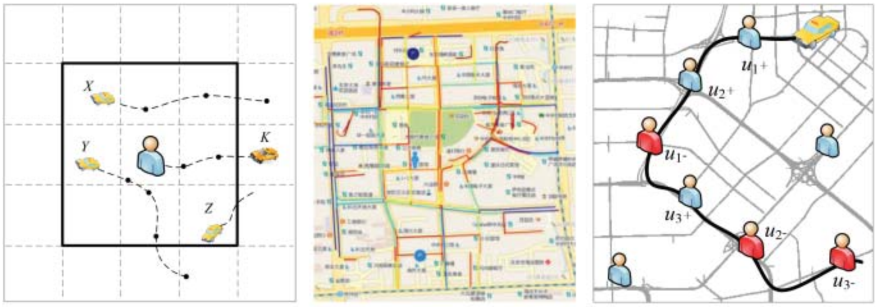
(b) Framework of T-Drive system

Fig. 6. T-Drive: driving directions based on taxi trajectories.

and the probability of accidents. So, the taxi trajectories imply traffic patterns and human intelligence. To deal with data sparsity (i.e., many road segments would not have taxis traversing), the citywide traffic patterns is modeled as a landmark graph, as shown in Figure 6(a), where red nodes are top-k road segments (titled landmarks) frequently traveled by taxis, and each blue edge denotes the aggregation of taxis' commutes between two landmarks. The travel time of each landmark edge is estimated based on the taxi data using a VE (variance and entropy) clustering algorithm. T-Drive uses a two-stage routing algorithm that first searches the landmark graph for a rough route (represented by a sequence of landmarks) and then connects these landmarks with a detailed route.

The second version of T-Drive [Yuan and Zheng et al. 2011b] mines taxis' historical trajectories and weather condition records to build four landmark graphs corresponding to different weather and days, as shown in Figure 6(b). The system also calculates the real-time traffic according to the recently received taxi trajectories and predicts future traffic conditions based on the real-time traffic and the corresponding landmark graph. A user submits a query, consisting of a source q_s, a destination q_d, a departure time $\underline{t}$, and a custom factor α, from a GPS-enabled mobile phone. Here, α is a vector representing how fast the user typically drives on different landmark edges. α is set by a default value at the very beginning and is gradually updated based on the trajectories the user has actually driven. T-Drive gives a much more accurate estimate for each user and will adjust its suggestions if a person's driving habits change over time. As a result, the system saves 5 minutes per 30-minute drive.

Wang et al. [2014] proposed a citywide and real-time model for estimating the travel time of any path (represented as a sequence of connected road segments) at the present time in a city, based on the GPS trajectories of vehicles received in present time slots and over a period of history as well as map data sources. The problem has three challenges. The first is the data sparsity problem; that is, many road segments may not be traveled by any GPS-equipped vehicles in the present time slot. In most cases, we cannot find a trajectory exactly traversing a query path either. Second, for the fragment of a path with trajectories, there are multiple ways of using (or combining) the trajectories to estimate the corresponding travel time. Finding an optimal combination is a challenging problem, subject to a tradeoff between the length of a path and the number of trajectories traversing the path (i.e., support). Third, we need to instantly answer users' queries, which may occur in any place of a city. This calls for an efficient, scalable, and effective solution that can enable a citywide and real-time travel time estimation. To address these challenges, Wang et al. modeled different drivers' travel times on



(a) Search for taxi dispatching (b) taxi-passenger recommendation (c) Dynamic taxi ridesharing

Fig. 7. Three categories of systems for improving taxi services.

different road segments in different time slots with a three-dimensional tensor. Combining with geospatial, temporal, and historical contexts learned from trajectories and map data, they filled in the tensor's missing values through a context-aware tensor decomposition approach. They then devised and proved an object function to model the aforementioned tradeoff, with which we find the most optimal concatenation of trajectories for an estimate through a dynamic programming solution. In addition, they proposed to use frequent trajectory patterns (mined from historical trajectories) to scale down the candidates of concatenation and a suffix-tree-based index to manage the trajectories received in the present time slot. The proposed solution was evaluated based on extensive experiments using GPS trajectories generated by more than 32,000 taxis over a period of 2 months. The results demonstrate the effectiveness, efficiency, and scalability of the method beyond baseline approaches, such as a simple summation of each individual road segment's travel time.

3.2.2. Improving Taxi Services. Taxis are an important commuting mode between public and private transportations, providing almost door-to-door traveling services. In major cities like New York City and Beijing, people usually wait for a nontrivial time before taking a vacant taxi, while taxi drivers are eager to find passengers. Effectively connecting passengers with vacant taxis is of great importance to saving people's waiting time, increasing taxi drivers' profit, and reducing unnecessary traffic and energy consumption. To address this issue, three categories of research have been done:

(1) Taxi dispatching systems: These kinds of systems [Lee et al. 2004] accept a user's booking request and assign taxis to pick up the user. Most systems request people to book a taxi in advance, therefore reducing the flexibility of taxi services. Some real-time dispatching systems search for proper taxis around a user based on the nearest neighbor principle of distance and time. The main challenge the system is faced with is the uncertainty of taxis' movement when searching for taxis [Phithakkitnukoon et al. 2010; Yamamoto et al. 2010]. As shown in Figure 7(a), taxi K may be a better candidate than (X, Y, Z) to pick up the user, if we could know taxi K is moving toward the user while others are leaving out the spatial range. In addition, the traffic condition on routes should also be considered to estimate the travel time to pick up the user.
(2) Taxi recommendation systems: This category of systems approaches the problem from the perspective of recommendation. Ge et al. [2010] developed a mobile recommender system, which has the ability to recommend a sequence of pick-up points for taxi drivers or a sequence of potential parking positions. The goal of the system is to maximize the probability of business success and reduce energy consumption. T-Finder [Yuan and Zheng et al. 2011a, 2014] provides taxi drivers with some

locations and the routes to these locations, toward which they are more likely to pick up passengers quickly (during the routes or in these locations) and maximize the profit of the next trip. T-Finder also suggests to people some locations (within walking distance) where they can easily find vacant taxis. As illustrated in Figure 7(b), the probabilities of finding a vacant taxi on different road segments are visualized with different colors, in which red means very difficult and blue denotes very likely. The parking places of taxis are also detected from the GPS trajectories of taxis, with an estimate of the number of taxis that will be arriving in the next half hour. The major challenge of this category of system is to deal with the data sparsity problem. For instance, how should the probability of finding a vacant taxi on road segments without sufficient data be calculated?

(3) Taxi ridesharing services: Taxi ridesharing is of great importance to saving energy consumption and easing traffic congestion while satisfying people's need in commute. T-Share [Ma and Zheng et al. 2013] is a large-scale dynamic taxi-sharing system that accepts passengers' real-time ride requests sent from smartphones and schedules taxis to pick up passengers via ridesharing, subject to time, capacity, and monetary constraints. As illustrated in Figure 7(c), a taxi is scheduled to sequentially pick up u_1 and u_2, drop off u_1, pick up u_3, and drop off u_2 and u_3, where + means a pick-up and − denotes a drop-off. T-Share maintains a spatiotemporal index that stores the status of each taxi, consisting of current location, number of passengers on board, and the planned route to deliver these passengers. When receiving a ride request, T-Share first searches the index for a set of candidate taxis that are likely to satisfy a user's query based on some temporal constraints. A scheduling algorithm is then proposed to insert the query's trip into the existing schedule of each candidate taxi, finding the taxi that satisfies the query with the minimum increase of travel distance. The system creates a win-win-win scenario, yielding significant social and environmental benefits. According to a simulation based on the taxi trajectories generated by over 30,000 taxis in Beijing, compared with traditional nonridesharing, the technology is able to save 120 million liters of gasoline per year in Beijing, which can support 1 million cars for 1.5 months, save $150 million US, and reduce 246 million KG of CO_2 emissions. In addition, passengers save 7% in taxi fare and have a 300% higher chance of being served, while the income of taxi drivers increases 10% [Ma and Zheng et al. 2013].

The difficulty of achieving such a taxi-sharing system lies in two aspects. One is to model the time, capacity, and monetary constraints for taxi trips. The other is the heavy computational load caused by the dynamics and large scale of passengers and taxis, which calls for efficient search and scheduling algorithms. Taxi users usually submit their queries last minute before a departure rather than scheduling in advance. A ride request can come from anywhere and at any time, while taxis are continually traveling around in a city. Of course, to push this technology into reality, there are still other nontechnical problems that need to be solved (e.g., the credit of a passenger and taxi driver as well as some security issues).

3.2.3. Improving Public Transportation Systems. By 2050, it is expected that 70% of the world's population will be living in cities. Municipal planners will face an increasingly urbanized and polluted world, with cities everywhere suffering an overly stressed road transportation network. Building more effective public transportation systems, as alternatives to private vehicles, has thus become an urgent priority, both to provide a good quality of life and a cleaner environment and to remain economically attractive to prospective investors and employees. Public mass transit systems, coupled with integrated fare management and advanced traveler information systems, are considered key enablers to better manage mobility. In the following subsections, we review some of the latest applications of urban computing across three public transport modalities: buses, subways, and shared bicycle schemes.

1) *Bus services*: In order to attract more riders, bus services need to be not only more frequent but also more reliable. Watkins et al. [2011] conducted a study on the impact of providing real-time bus arrival information directly on riders' mobile phones and found it to reduce not only the perceived wait time of those already at a bus stop but also the actual wait time experienced by customers who plan their journey using such information. In other words, mobile real-time information has the ability to improve the experience of transit riders by making the information available to them before they reach the stop. In cases where GPS receivers have not been deployed on buses themselves, alternative solutions have been explored to gather the same information, but in a cheaper and less intrusive manner. Zimmerman et al. [2011] were the first to develop, deploy, and evaluate a system called Tiramisu, where commuters share GPS traces, as collected from the GPS receivers on their mobile phones. Tiramisu then processes incoming traces and generates real-time arrival time predictions for buses. As the GPS trajectories may be a mixture of different transportation modes (e.g., first taking a bus and then walking), Zheng et al. [2008a, 2008b, 2010b] proposed a method to infer a user's transportation modes (consisting of driving, walking, riding a bike, and taking a bus) in each segment of a trajectory. Once the trajectories have been classified by transportation modes, a more accurate estimate can be made for bus travel time or driving time prediction.

   As the process of urbanization keeps changing our cities, it is essential for bus transit services to adapt their routes over time to keep meeting the mobility demands of their citizens. However, the pace at which bus routes are updated is much slower than the pace at which citizens' needs change. Bastani et al. [2011] proposed a data-centric approach to tackle the issue: they developed a new mini-shuttle transit system called flexi, whose routes are flexibly derived from actual demand by analyzing passenger trip data from a large set of taxi trajectories. In a similar vein, Berlingerio et al. [2013] analyzed the anonymized and aggregated CDRs from Abidjan in the Ivory Coast, with the aim to inform the planning of a public transit network using mobile phones. In this context, the resource-intensive transportation planning processes prevalent in the West are not affordable; using mobile phone data to perform transit analysis and optimization represents a new frontier for transport planning in developing countries, where mobile phones have deep penetration so that their anonymized flow data can be readily mined.

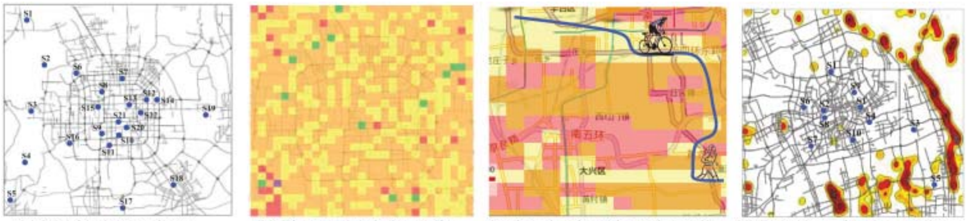
2) *Subway services*: Automated fare collection (AFC) systems (e.g., London's Oyster Card, Seattle's Orca, Beijing's Yikatong, Hong Kong's Octopus, etc.) have been introduced and are now widely adopted in many metropolitan cities around the world. Apart from simplifying access to the city subway network of train services, these smart cards create a digital record every time a trip is made, which can be linked back to the individual traveler. Mining the travel data that is created as travelers enter and exit stations can give vast insight into the travelers themselves: their implicit preferences, travel times, and commuting habits.

   Lathia et al. [2010] mined AFC data with the aim to build more accurate travel route planners. They used data collected from the London Underground (tube) system, which implements electronic ticketing in the form of RFID-based contactless smart cards (Oyster cards). Unlike some AFC systems, Oyster cards must be used both when entering and when exiting stations. An in-depth analysis of two large datasets of the London's tube usage demonstrated that there are substantial differences between travelers that emerge. Based on the insights, they have automatically extracted features from the AFC data that implicitly capture information about a user's familiarity with a journey, a user's similarity to other travelers, and a user's journey context. Finally, they have used these features to develop personalized travel tools whose aims can be formalized as prediction problems: (1) predicting personalized travel times between any origin and destination pairs to provide users with accurate estimates of their transit time and (2) predicting and ranking the interest that individual travelers will have in receiving alert notifications about particular stations based on their past travel histories. In a follow-up work, Ceapa et al. [2012] performed a spatiotemporal analysis of the same historical Oyster card traces and discovered that crowdedness is a highly regular phenomenon during the working week, with spikes occurring for rather short time intervals. They went on to build predictors of crowding levels, which could then be incorporated in advanced traveler information systems to offer travelers more personalized and quality-based planning services.

Lathia and Capra [2011a] also analyzed AFC data to estimate future travel habits, once again for the case of London. By analyzing historical travel traces, they have been able to extract features about when, where, and how often an individual travels, which can then be predicted with a high level of accuracy. They have leveraged these findings to build tools that can recommend to travelers what is the best fare for them to purchase, based on their expected travel habits. In so doing, they have demonstrated they could offer savings of several hundreds of thousands of pounds a year, as misconceptions about our own travel behaviors often lead to the incorrect fare being purchased [Lathia and Capra 2011b].

3) *Bike-sharing systems*: As the world population grows and an ever-increasing proportion of people live in cities, designing, maintaining, and promoting sustainable urban mobility modes are becoming of paramount importance. Shared bicycle schemes [Shaheen et al. 2010] are one such example: their proliferation throughout the world's metropolises clearly reflects the belief that providing easy access to healthy (and quick) modes of transport will lead cities away from the congestion and pollution problems they currently face. Detailed records are often available about shared bikes' movement (from where/when a bike was taken to where/when a bike was returned), thus allowing researchers to analyze these digital traces to help end-users, who may benefit from both understanding and forecasting how the system will be used when planning their own trips; transport operators, who may benefit from more accurate models of bicycle flows in order to appropriately load-balance the stations throughout the day; and urban planners, who can leverage flow data when designing social spaces and policy interventions.

Froehlich et al. [2009] were among the first to take a data-centric approach to shared bicycle systems by applying a host of data-mining techniques to uncover spatiotemporal trends in a city's data. They performed an in-depth analysis of 13 weeks of Barcelona's Bicing system (Spain), clearly demonstrating the relationships between time of day, geography (particularly, clusters of stations within geographic areas of the city), and usage. Kaltenbrunner et al. [2010] performed a similar study of Bicing in Barcelona, and Borgnat et al. performed one of Lyon, France [2009]. In these studies, the authors focus on temporal properties of the bicycle station data in order to train and test classifiers that predict the state (availability of bicycles) of each station. Nair et al. [2013] analyzed data from Paris's (France) Velib', relating usage to rail station proximity: they uncover the relation between bicycle usage and multimodal trips, thus providing key insights into station placement policy. Finally, Lathia et al. [2012] analyzed the London's Cycle Hire scheme over two different 3-month periods and derived quantitative evidence of how an access policy change impacted bike usage across the whole city.



(a) Monitoring stations (b) Fine-grained air quality (c) Routing based on air quality (d) Locations for new stations

Fig. 8. Monitoring real-time and fine-grained air quality using big data.

3.3. Urban Computing for the Environment

Without effective and adaptive planning, urbanization's rapid progress will become a potential threat to cities' environment. Recently, we have witnessed an increasing trend of pollution in different aspects of the environment, such as air quality, noise, and rubbish, around the world. Protecting the environment while modernizing people's lives is of paramount importance in urban computing.

3.3.1. Air Quality. Information about urban air quality (e.g., the concentration of $PM_{2.5}$) is of great importance to protecting human health and controlling air pollution. Many cities are monitoring $PM_{2.5}$ by building ground-based air quality measurement stations. However, there are only a limited number of air quality measurement stations in a city (as illustrated in Figure 8(a)) due to the expensive cost of building and maintaining such a station. Unfortunately, air quality varies by locations nonlinearly and depends on multiple factors, such as meteorology, traffic volume, and land uses. As a result, we do not really know the air quality of a location without a measurement station.

The advances in mobile communication and sensing technologies have proliferated the crowdsourcing-based applications, which decompose a complex problem into small tasks and distribute these small tasks to a network of users. The returns from individual users will formulate collective knowledge that can solve the complex problem. *Copenhagen Wheels* is a project that installs environmental sensors in a bike's wheel to sense the fine-grained environmental data of a city, including temperature, humidity, and the concentration of CO_2. The human labor for riding a bike is transferred into the power to support the operation of on-bike sensors. In addition, the wheel can communicate with a user's mobile phone, through which the collected information is sent to a backend system. Likewise, Devarakonda et al. [2013] presented a vehicular-based approach for measuring fine-grained air quality in real time. They devised a mobile device, consisting of a GPS receiver, CO sensor, and cellular modem. Installing such a device on multiple vehicles, they would be able to monitor the concentration of CO throughout a city. Though having a huge potential, monitoring the environment through crowdsourcing only works well for a few gasses, such as CO_2 and CO. The device for measuring aerosol, like $PM_{2.5}$ and PM_{10}, are not easily portable for an individual. Moreover, these devices need a relatively long period of sensing time (e.g., 2 hours) before generating an accurate measurement.

Another branch of research (e.g., [Guehnemann et al. 2004]) is to first estimate the traffic flow on road surfaces based on floating car data and then calculate the emission of vehicles based on some empirical equations formulated by environmentalists. This is a promising approach to estimate the air pollution nearby to roads but cannot reveal the air quality of an entire city as the emission from vehicles is only a part of air pollution.

Different from existing solutions, Zheng et al. [2013b, 2014a] inferred the real-time and fine-grained air quality information throughout a city (as demonstrated in Figure 8(b)) based on the (historical and real-time) air quality data reported by existing monitor stations and a variety of data sources observed in the city, such as meteorology, traffic flow, human mobility, structure of road networks, and POIs. Instead of using classical physical models that explicitly combine factors in a formula based on empirical assumptions, they approach this problem from a big data perspective, that is, using data-mining and machine-learning techniques to build a network between a diversity of data sources and air quality indexes (see more technique details in Section 4.3). The fine-grained air quality information could help people figure out, say, when and where to go jogging, or when they should shut the window or put on a facemask, as depicted in Figure 8(c). The information can also be used to suggest the location where we might need to build additional monitoring stations if current stations are not enough, as shown in Figure 8(d). This is also a step toward identifying the root cause of air pollution in a city, therefore informing government's decision making. The approach was evaluated with real data sources obtained in 10 cities, including Beijing, Shanghai, Wuhan, and Shenzhen. A public website is available at http://urbanair.msra.cn/.

Chen et al. [2014] introduce an indoor air quality monitoring system deployed in four Microsoft Campuses in China. The system is composed of sensors deployed on different floors of a building, a cloud collecting and analyzing the data from the sensors and the public air pollution information, and clients that display real-time air quality data of booth outdoor and indoor environments to end-users. The system provides users with indoor air quality information that can inform people's decision making in office areas, such as when to work out in a gym or turn on an additional air filter in an office. The gap between the concentration of $PM_{2.5}$ in outdoor and indoor environments can measure the effectiveness of an HVAC (heating, ventilation, and air conditioning) system in filtering $PM_{2.5}$. In addition, the system integrates outdoor air quality information with indoor measurements to adaptively control HVAC settings with a view on optimizing runtimes w.r.t. energy efficiency and air quality conservation. Using a neural network-based approach, the system can even predict the purification time that HVAC needs to reduce the concentration of indoor $PM_{2.5}$ to below a healthy threshold, based on six factors, such as the concentration of outdoor/indoor $PM_{2.5}$, barometer pressure, and humidity. Given the purification time and the timing that people start working in a building, the number of hours that a HVAC system should be turned on ahead of its original schedule can be suggested. Extensive experiments using 3-month data demonstrate the advantage of our approach beyond baseline methods, (e.g., linear regression). With a minor decrease in accuracy, the system can infer a shorter purification time, thus saving a lot of energy.

3.3.2. Noise Pollution. The compound functions of a city and its complex settings that incorporate different infrastructures and millions of people inevitably generate a lot environmental noise. As a result, a large number of people around the world are exposed to high levels of noise pollution, which can cause serious illnesses ranging from hearing impairment to negatively influencing productivity and social behavior [Rana et al. 2010].

As an abatement strategy, a number of countries, such as the United States, the United Kingdom, and Germany, have started monitoring noise pollution. They typically use a noise map (a visual representation of the noise level of an area) to assess noise pollution levels. The noise map is computed using simulations based on inputs such as traffic flow data, road or rail type, and vehicle type. Since the collection of such input data is very expensive, these maps can be updated only after a long period of time.

Silvia et al. [2008] assess environmental noise pollution in urban areas by using wireless sensor networks. However, deploying and maintaining a citywide sensor network, especially in major cities like New York City, is very expensive, in terms of money and human resources.

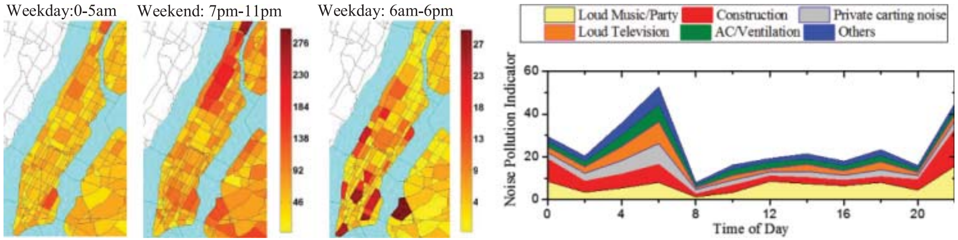
Another solution is to take advantage of crowdsourcing, where people collect and share their ambient environmental information using a mobile device (e.g., a smartphone). For example, NoiseTube [Nicolas et al. 2009] presents a person-centric approach that leverages the noise measurements shared by mobile phone users to paint a noise map in a city. Based on NoiseTube, D'Hondt and Stevens [2011] conducted a citizen science experiment for noise mapping a 1-km^2 area in the city of Antwerp. Extensive calibration experiments were also carried out investigating both frequency-dependent and white noise behavior. The main objective of this experiment is to investigate the quality of the obtained noise map by participatory sensing, compared with official simulation-based noise maps.

In Rana et al. [2010], an end-to-end, context-aware, noise mapping system called Ear-Phone is designed and implemented. Different from phone users actively uploading their measurements in Nicolas et al. [2009] and D'Hondt and Stevens [2011], an opportunistic sensing approach is proposed, where noise-level data is collected without informing smartphone users. One major problem solved in this paper is classifying the phone sensing context, that is, in pocket (bag) or hand, which is related with the accuracy of the sensed data. To recover a noise map from incomplete and random samples, Rana et al. [2013] further study a number of different interpolation and regularization methods, including linear interpolation, nearest neighbor interpolation, Gaussian process interpolation, and L1-norm minimization methods.

Modeling citywide noise pollution is actually much more than just measuring the intensity of noises, as the measurement of noise pollution also depends on people's tolerance to noises, which changes over time of day. For example, in the night, people's tolerance to noise is much lower than in the daytime. A less loud noise in the night may be nevertheless considered a heavier noise pollution. Consequently, even if we could deploy sound sensors everywhere, diagnosing urban noise pollution solely based on sensor data is not enough. Furthermore, urban noises are usually a mixture of multiple sound sources. Understanding the composition of noises (e.g., in the evening rush hours, 40% of noises in a place are from pub music, 30% from vehicle traffic, and 10% from construction) is vital to help tackle noise pollution.

Since 2001, New York City has opened a platform titled 311 to allow people to complain about imperfections of the city by using a mobile app or making a phone call; noise is the third largest category of complaints in the 311 data. As each complaint about noises is associated with a location, timestamp, and fine-grained noise category, such as loud music or construction noises, the data is actually a result of "human as a sensor" and "crowd sensing," containing rich human intelligence that can help diagnose urban noises. Zheng et al. [2014b] infer the fine-grained noise situation (consisting of a noise pollution indicator and the composition of noises) of different times of day for each region of New York City by using the 311 complaint data together with social media, road network data, and POIs. According to the overall noise pollution indicator, we can rank locations in different time spans (e.g., 0 AM to 5 AM weekends and 7 PM to 11 PM weekends), as illustrated in Figure 9(a); the darker the area is, the heavier the noise pollution is. Or we can rank locations by a particular noise category like *construction*, as depicted in Figure 9(b). We can also check the noise composition of a particular location changing over time (e.g., Time Square), as shown in Figure 9(c).

They model the noise situation of New York City with a three-dimensional tensor, where the three dimensions stand for regions, noise categories, and time slots. By filling in the missing entries of the tensor through a context-aware tensor decomposition approach, they recover the noise situation throughout New York City. The information of noise can not only facilitate an individual's quality of life (e.g., help find a quiet place to settle down) but also inform governmental officials' decision making on tackling noise pollution.



(a) Overall noises (b) Construction (c) Noise of different categories in Time Square

Fig. 9. Diagnosing the noise pollution of New York City.

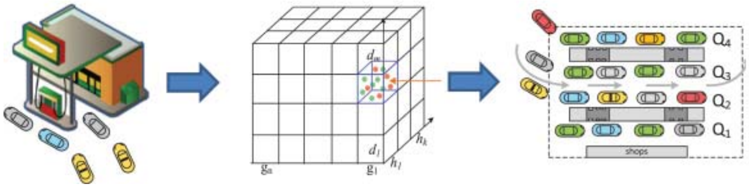


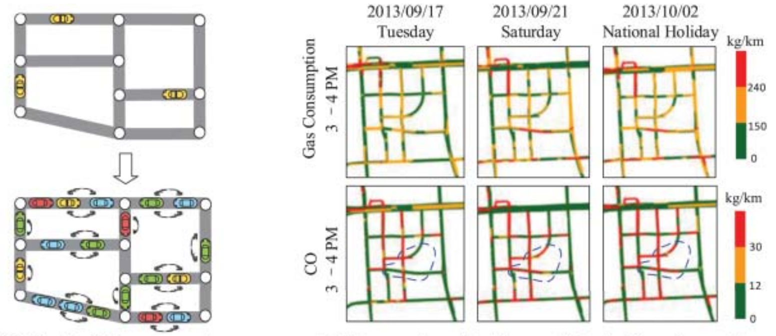
Fig. 10. Crowdsensing urban refueling behavior with GPS-equipped taxis.

3.4. Urban Computing for Urban Energy Consumption

The rapid progress of urbanization is consuming more and more energy, calling for technologies that can sense city-scale energy cost, improve energy infrastructures, and finally reduce energy consumption.

3.4.1. Gas Consumption. Zhang et al. [2013] proposed a step toward real-time sensing of refueling behavior and citywide petrol consumption. The method uses a "human as a sensor" approach by analyzing and drawing inferences from GPS trajectories passively collected by taxicabs. At first, they detect the refueling events, which are visits by taxis to gas stations, from the GPS trajectories, as illustrated in the left part of Figure 10. The detection of refueling events includes the time spent waiting at the gas station and the time spent refueling the vehicle. Second, as shown in the middle part of Figure 10, they build a tensor with the three dimensions respectively denoting gas stations, day of the week, and time of day. Each entry contains the refueling events detected at a particular time slot in a particular day and in a particular gas station. For entries that cover enough detected refueling events, the time spent in each of these entries is estimated from the distribution of the refueling events. For those with few or even without refueling events, they use a context-aware collaborative filtering approach to solve the data sparsity problem. Finally, as depicted in the right part of Figure 10, they treat each gas station as a queue system, and time spent in the station is used to calculate drivers' arrival rate, which is the number of customers during this period and can indicate the petrol consumption indirectly. Therefore, the output is a global estimate of time spent and fuel use at each gas station in each time period. Refer to Example 3 in Section 4.3 for more details of the methodology.

Eco-feedback technologies that provide information on the driving behavior have shown to be an effective means to stimulate changes in driving in favor of energy conservation and emission reduction. Tulusan et al. [2012] demonstrated that a smartphone application can improve fuel efficiency even under conditions where monetary incentives are not given (i.e., where the drivers do not pay for fuel). Given the large share of corporate cars, findings are also of high practical importance and motivate future research on eco-driving feedback technologies. Recently, using monetary incentives, some insurance companies have been encouraging customers to share driving



(a) Goal of the research (b) Some visualizations of the inferred results

Fig. 11. Inferring gas consumption and pollution emission from vehicles based on sparse trajectories.

behaviors recorded by a variety of car-sensor data, like pushing a gas pedal and brake and making a direction turn. The data can be used to estimate a driver's probability to encounter an accident, therefore helping determine the price for next year's insurance fare. Having such kind of detailed driving behavior data of a large number of people will enable us to understand the instant energy consumption of a city and analyze the energy cost of a particular route, therefore coming up with some solution for energy conservation (e.g., suggesting the route using the least amount of gas).

Shang et al. [2014] instantly inferred the gas consumption and pollution emission of vehicles traveling on a city's road network in the current time slot using GPS trajectories from a sample of vehicles (e.g., taxicabs), as illustrated in the Figure 11(a). The knowledge can be used not only to suggest cost-efficient driving routes but also to identify the road segments where gas has been wasted significantly. In the meantime, the instant estimation of the pollution emission from vehicles can enable pollution alerts, and, in the long run, help diagnose the root cause of air pollution.

They first computed the travel speed of each road segment using the GPS trajectories received recently. As many road segments are not traversed by trajectories (i.e., data sparsity), a travel speed estimation model is proposed based on a context-aware matrix factorization approach. The model leverages features learned from other data sources (e.g., map data and historical trajectories) to deal with the data sparsity problem. A Traffic Volume Inference model (TVI) was then proposed to infer the number of vehicles passing each road segment per minute. TVI is an unsupervised dynamic Bayesian network that incorporates multiple factors, such as the travel speed, weather conditions, and geographical features of a road. Given the travel speed and traffic volume of a road segment, the gas consumption and emission can be calculated based on existing environmental theories.

Figure 11(b) demonstrates the gas consumption and emission of NO_x around the Zhongguancun area, which is a place with many companies and entertainment areas, during 3 days. In the time slot from 3 PM to 4 PM, the time before evening the rush hour, this area has less gas consumption in the workday than in the weekend and holiday, because people are still working indoors. When the time goes to weekends and holidays, many people travel to this region for the purpose of entertainment (e.g., to go shopping and watch a movie), leading to more energy consumption and emission of CO, denoted by red segments. There is a movie theater, a supermarket, and two shopping centers located in the region marked by the broken curve.

3.4.2. Electricity Consumption. Efficient integration of energy from renewable sources and meeting the increased demand resulting from an increase in electrification of vehicles and heating are key to the sustainability of the electricity supply in order to optimize residential energy usage. Intelligent demand response mechanisms are needed to shift energy usage to periods of low demand or to periods of high availability of renewable energy. Intelligent algorithms, implemented at either the device level or the community/transformer level, enable devices to meet individual device and user policies as well as stay within community-assigned energy usage limits.

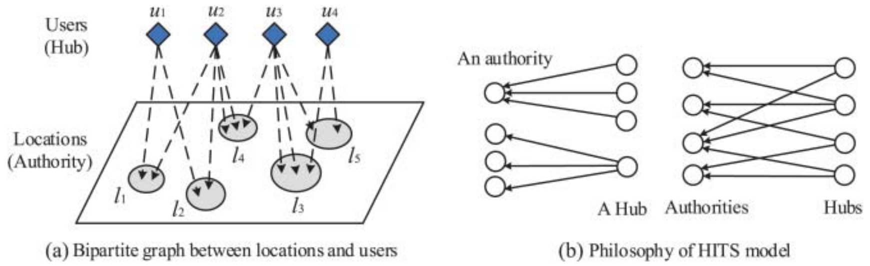
In Dusparic et al. [2013], each electric vehicle within a community is controlled by a reinforcement learning agent, further supported by a short-term load prediction algorithm [Marinescu et al. 2014]. Each agent's local goals are to minimize the charging price (which is dynamic and directly proportional to current energy demand) and to meet desired user utility (e.g., have an EV's battery 80% charged in time for departure). Each agent also has a goal to keep the community transformer level under a target limit (by minimizing, for example, the number of vehicles charging during peak periods). Demand is dynamically repredicted if real-time monitoring shows deviations of actual demand from predicted demand. Galvan-Lopez et al. [2014] propose an alternative approach, where instead of each vehicle agent making its own decisions, a globally optimal charging schedule evolves using genetic algorithms and is communicated to the electric vehicles. In Harris et al. [2014], intelligent set point control algorithms at the transformer level send out signals to controllable devices (e.g., EVs or water heaters) indicating either a probability they should use to determine whether they should be charging/on at any particular point or the degree to which each device's variable power chargers should be turned on. This enables fine-grained control of device demand to fill the gaps between uncontrollable electric load and the target transformer load to smooth out overall energy demand.

Momtazpour et al. [2013] presented a framework to support charging and storage infrastructure design for electric vehicles. A coordinated clustering technique was proposed to work with network models of urban environments to aid in placement of charging stations for an electrical vehicle deployment scenario. Issues that have been taken into account include (1) prediction of EV charging needs based on owners' activities, (2) prediction of EV charging demands at different locations in the city and available charge of EV batteries, (3) design of distributed mechanisms that manage the movements of EVs to different charging stations, and (4) optimization of the charging cycles of EVs to satisfy users' requirements while maximizing vehicle-to-grid profits.

3.5. Urban Computing for Social Applications

Although there are already many social networking services on the Internet, in this section, we focus on introducing location-based social networks (LBSNs), which are formally defined as follows in Zheng [2011a, 2012a]:

A location-based social network (LBSN) not only means adding a location to an existing social network so that people in the social structure can share location-embedded information but also consists of the new social structure made up of individuals connected by the interdependency derived from their locations in the physical world as well as their location-tagged media content, such as photos, video, and texts. Here, the physical location consists of the instant location of an individual at a given timestamp and the location history that an individual has accumulated in a certain period. Further, the interdependency includes not only that two persons co-occur in the same physical location or share similar location histories but also the knowledge (e.g., common interests, behavior, and activities) inferred from an individual's location (history) and location-tagged data.



(a) Bipartite graph between locations and users

(b) Philosophy of HITS model

Fig. 12. Inferring the most interesting places and most experienced users.

LBSNs bridge the gap between users' behavior in digital and physical worlds [Cranshaw 2010], which well matches the nature of urban computing (as presented in Section 2.3). In a location-based social network, people can not only track and share the location-related information of an individual but also leverage collaborative social knowledge learned from user-generated and location-related content, such as check-ins, GPS trajectories, and geo-tagged photos [Zheng et al. 2011c, 2012b]. Examples of LBSNs include the widely used Foursquare and a research prototype called GeoLife [Zheng et al. 2008c, 2008d, 2009a, 2010d]. With LBSNs, we can understand users and locations and explore the relationship between them:

1) *Estimate user similarity*: An individual's location history in the real world implies, to some extent, his or her interests and behaviors. Accordingly, people who share similar location histories are likely to have common interests and behaviors. The similarity between users inferred from their location histories can enable friend recommendations [Li et al. 2008], which connect users with similar interests even when they may not have known each other previously, and community discovery [Hung et al. 2009], which identifies a group of people sharing common interests. To better estimate the similarity between users, more information, such as the visiting sequences between locations, the geospatial granularity of a location, and the popularity of a location, are considered in Zheng et al. [2010d]. In addition, in order to be able to calculate the similarity of users living in different cities (i.e., having little geospatial overlaps in users' location histories), Xiao et al. [2010, 2012] extended Zheng's research from physical locations into the semantic space of locations by considering the categories of points of interests in the location visited by a user.
2) *Finding local experts in a region* [Zheng et al. 2009c]: With users' locations, we are able to identify the local experts who have richer knowledge about a region (or a topic like shopping) than others. Their travel experiences (e.g., the locations where they have been) are more accountable and valuable for travel recommendations. For instance, local experts are more likely to know about high-quality restaurants than some tourists.
3) *Location recommendations*: Finding the most interesting locations in a city is a general task that a tourist wants to fulfill when traveling to an unfamiliar city [Zheng et al. 2009c]. However, the interest level of a location depends not only on the number of people who have visited the location but also these people's travel knowledge. For example, the most frequently visited location in a city could be its railway station or airport, which might not be an interesting location recommendation. On the contrary, some locations that attract experienced people (i.e., with rich travel knowledge) may be truly interesting. The problem is then how to determine an individual's travel experience. As shown in Figure 12(a), Zheng et al. [2009c] formulated a bipartite graph between users and locations and employed a HITS

> (hypertext-induced topic search)-based model to infer the interest level of a location and the travel knowledge of a user (as illustrated in Figure 12(b)). The general idea is that users' travel experiences and the interest level of a location have a mutual reinforcement relationship. More specifically, a user's knowledge can be represented by the summation of the interests of the locations the user has visited; in turn, the interest of a location is represented by the summation of the knowledge of the users who have visited this location.

In some scenarios, we can consider a user's preferences (e.g., Italian food and watching movies) and contexts (e.g., current location and time) when suggesting location recommendations [Ye et al. 2011; Liu et al. 2013]. One simple method is to formulate a user–location matrix where each row denotes a user, each column denotes a location, and each entry stands for the number of visits of a particular user in a particular location. Then some collaborative filtering methods can be used to fill in the entries without a value. This kind of method calculates the similarity between users solely based on the two rows denoting the two users' location histories but misses useful information, such as the aforementioned visiting sequences between locations. Considering rich information, Zheng et al. [2010e] incorporated the user similarity they inferred in a paper [Li et al. 2008] into a user-based CF model to infer the missing value in the user–location matrix. Though having a deeper understanding of user similarity, the method suffers from the increasing scale of users since the model needs to calculate the similarity between each pair of users. To address this issue, location-based collaborative filtering was proposed in Zheng et al. [2011d]. This model computes the correlation between locations based on the location histories of the users visiting these locations [Zheng et al. 2009b]. The correlation was then used as a kind of similarity between locations in an item-based CF model. Given the limited geographical space (i.e., the number of locations is limited), this location-based model is more practical for a real system.

As a user can only visit a limited number of locations, the user–location matrix is very sparse, leading to a big challenge to traditional collaborative filtering-based location recommender systems. The problem becomes even more challenging when people travel to a new city they have not visited. To this end, Bao et al. [2012] presented a location-based and preference-aware recommender system that offers a particular user a set of venues (such as restaurants and shopping malls) within a geospatial range with the consideration of both user personal preferences, which are automatically learned from location history, and (2) social opinions, which are mined from the location histories of the local experts. This recommender system can facilitate people's travel not only near their living areas but also to a city that is new to them.

Itinerary planning: Sometimes a user needs a sophisticated itinerary conditioned by the user's travel duration and departure place. The itinerary could include not only stand-alone locations but also detailed routes connecting these locations and a proper schedule, for example, the typical time of day that most people reach the location and the appropriate time length that a tourist should stay there. Yoon et al. [2010, 2011] planned a trip in terms of the collective knowledge learned from many people's GPS trajectories. Wei et al. [2012] learned the most likely route traveling between two query points through learning from many check-in data.

Location–activity recommender: This recommender provides a user with two types of recommendations: (1) the most popular activities that can be performed in a given location and (2) the most popular locations for conducting a given activity, such as shopping. These two categories of recommendations can be mined from a large number of users' trajectories and location-tagged comments. To achieve the two goals, Zheng et al. [2010f] proposed a context-aware collaborative filtering model, which was solved by a matrix factorization method (refer to Section 4.4.2 for details). Furthermore, Zheng et al. [2010a, 2012d] extended the location–activity matrix into a tensor by considering users as the third dimension. By applying a context-aware tensor decomposition method, a personalized location–activity recommendation was proposed (Section 4.4.3 offers details).

Life patterns and styles understanding: The social media data, especially the geotagged tweets, photos, and check-ins, can help understand not only an individual's life patterns [Ye et al. 2009] but also a city's dynamics [Cranshaw 2012], topics [Yin et al. 2011], behavior patterns [Wakamiya et al. 2012], or lifestyles [Yuan et al. 2013a] when used aggregately. We can also compute the similarity between two cities according to the social media generated in the cities.

More details about LBSNs can be found in Zheng [2011a, 2011e] and a survey on recommendations in LBSNs in Bao et al. [2014]

3.6. Urban Computing for Economy

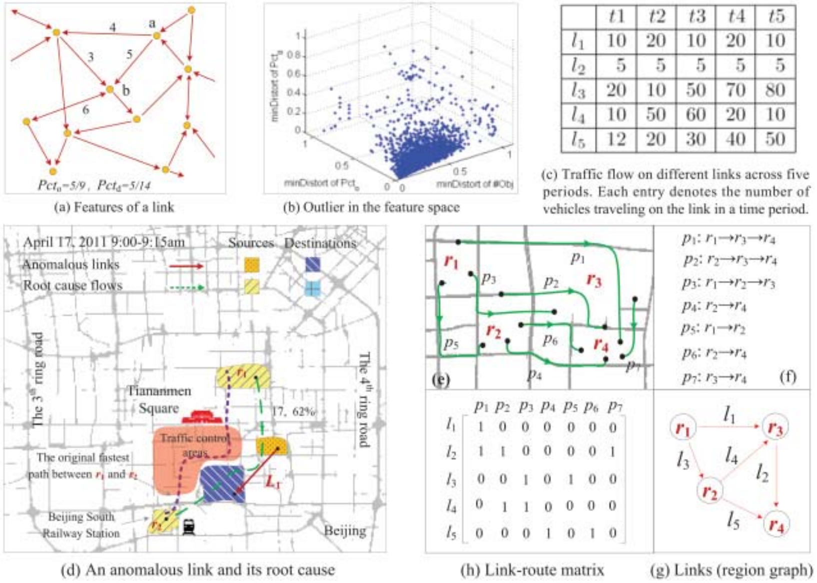
The dynamics of a city (e.g., human mobility and the number of changes in a POI category) may indicate the trend of the city's economy. For instance, the number of movie theaters in Beijing kept increasing from 2008 to 2012, reaching 260. This could mean that more and more people living in Beijing would like to watch a movie in a movie theater. On the contrary, some category of POIs is going to vanish in a city, denoting the downturn of the business. Likewise, human mobility could indicate the unemployment rate of some major cities, therefore helping predict the trend of a stock market.

Human mobility combined with POIs can also help the placement of some businesses. Karamshuk et al. [2013] studied the problem of optimal retail store placement in the context of location-based social networks. They collected human mobility data from Foursquare and analyzed it to understand how the popularity of three retail store chains in New York is shaped in terms of number of check-ins. A diverse set of data-mining features were evaluated, modeling spatial and semantic information about places and patterns of user movements in the surrounding area. As a result, among those features, the presence of user attractors (i.e., train station or airport) as well as retail stores of the same type to the target chain (i.e., coffee shop or restaurant) encoding the local commercial competition of an area, are the strongest indicators of popularity.

Combing more data sources, we can even predict the ranking of real estate. Fu et al. [2014] predicted the ranking of residential real estate in a city at a future time according to their potential *values* inferred from a variety of data sources, such as human mobility data and urban geography, currently observed around the real estate. Here, "value" means the ability to increase more quickly in a rising market and decrease more slowly than others in a falling market, quantified by discretizing the increasing or decreasing percentage over its previous price into five levels (R_1–R_5), where R_1 stands for the best and R_5 denotes the worst. The rank is of great importance to people when settling down or allocating capital investment. The problem is difficult, however, as the ranking depends on many factors, which vary in location nonlinearly and may even change over time. To address this issue, we first identify a set of discriminative features for each house by mining the geographic data (e.g., road networks and POIs) and traffic data around it. We then train a pair-wise learning-to-rank model by feeding a list of features–ranking pairs into an artificial neural network. A metric learning algorithm is also applied to identify the top 10 most influential features affecting the ranking, implicitly revealing the important factors determining the value of real estate.

3.7. Urban Computing for Public Safety and Security

Large events, pandemics, severe accidents, environmental disasters, and terrorism attacks pose additional threats to public security and order. The wide availability of



(a) Features of a link

(b) Outlier in the feature space

	$t1$	$t2$	$t3$	$t4$	$t5$
l_1	10	20	10	20	10
l_2	5	5	5	5	5
l_3	20	10	50	70	80
l_4	10	50	60	20	10
l_5	12	20	30	40	50

(c) Traffic flow on different links across five periods. Each entry denotes the number of vehicles traveling on the link in a time period.

(d) An anomalous link and its root cause

p_1: $r_1 \to r_3 \to r_4$
p_2: $r_2 \to r_3 \to r_4$
p_3: $r_1 \to r_2 \to r_3$
p_4: $r_2 \to r_4$
p_5: $r_1 \to r_2$
p_6: $r_2 \to r_4$
p_7: $r_3 \to r_4$ (f)

$$\begin{array}{c|ccccccc} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 \\ \hline l_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ l_2 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ l_3 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ l_4 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ l_5 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}$$

(h) Link-route matrix

(g) Links (region graph)

Fig. 13. Detecting anomalies from urban traffic based on distance.

different kinds of urban data provides us with the ability, on one hand, to learn from history how to handle the aforementioned threats correctly and, on the other hand, to detect them in a timely manner or even predict them in advance.

3.7.1. Detecting Traffic Anomalies. Traffic anomalies in urban areas could be caused by accidents, traffic control, protests, sports, celebrations, disasters, and other events. The detection of traffic anomalies can help disperse congestions, diagnose unexpected events, and facilitate people's drive. According to the surveys on anomaly detection [Chandola 2009; Hodge 2004], there are four major categories of methods: classification based, clustering based, distance based, and statistical based. In this article, we only introduce the latter two categories in the setting of urban traffic.

(1) *Distance-based methods*: This category of methods represents a spatial object (e.g., a region, a road segment, or a link connecting two regions) by a set of features (extracted from traffic data), which is then used to calculate the distance between two spatial objects. The spatial objects with a farther distance to others are considered outliers. Some thresholds, like three times the standard deviation, are usually employed to identify outliers.

Liu et al. [2011] partitioned a city into disjointed regions with major roads and gleaned the anomalous links between two regions according to the traffic of vehicles traveling between the two regions. They divided time of day into time bins and identified for each link three features, consisting of number of vehicles traveling the link in a time bin (#Obj), proportion of the these vehicles among all vehicles moving into the destination region (Pct_d), and that moving out of the origin region (Pct_o). As shown in Figure 13(a), regarding link $a \to b$, #$Obj = 5$, $Pct_d = 5/14$, and $Pct_o = 5/9$. The three features of a time bin were respectively compared with those in the equivalent time bins of previous days to calculate the minimum distort of each feature (i.e., minDistort #Obj, minDistort Pct_d, and minDistort Pct_o). Then, the link of the time bin can be represented in a three-dimensional space, with each dimension denoting the minimum distort of a feature, as depicted in Figure 13(b).

To normalize the effect of variances along different directions, the Mahalanobis distance was used to measure the most extreme points, which were regarded as outliers.

Following the aforementioned research, Sanjay et al. [2012] proposed a two-step mining and optimization framework to detect traffic anomalies between two regions and explain an anomaly with the traffic flows passing through the two regions. As illustrated in Figure 13(d), an anomalous link L_1 was found between two regions. However, the problem may not lie in the two regions. On April 17, 2011, traffic in Beijing had been diverted away from Tiananmen Square because of the Beijing marathon. Thus, the normal traffic route (shown as the dotted path) from region r_1 to the Beijing South Railway Station in r_2 was diverted and the dashed (green) path witnessed excess traffic. In short, the traffic flow on the green path leads to the anomaly. In the methodology, given a link matrix like that shown in Figure 13(c), they first used a PCA (principal component analysis) algorithm to detect some anomalous links, which were represented by a column vector b, with 1 denoting an anomaly detected on the link. An adjacent link–route matrix A was formulated based on the trajectories of vehicles, as illustrated from Figure 13(d) to 13(g). Each entry of the matrix denotes whether a route passes a link; 1 means yes; 0 means no. For instance, route p_1 passes l_1 and l_2. Then, the relationship between anomalous links and routes was captured by solving the equation $Ax = b$, where x is a column vector denoting which paths contribute to the emergency of these anomalies shown in b. Using L_1 optimization techniques, the x can be inferred.

Pan et al. [2013] identified traffic anomalies according to drivers' routing behavior on an urban road network. Here, a detected anomaly is represented by a subgraph of a road network, where drivers' routing behaviors significantly differ from their original patterns. They then try to describe the detected anomaly by mining representative terms from the social media that people posted when the anomaly happened. The system for detecting such traffic anomalies can benefit both drivers and transportation authorities, for example, by notifying drivers approaching an anomaly and suggesting alternative routes, as well as supporting traffic jam diagnosis and dispersal.

2) *Statistical-based methods*: The underlying principle of a statistical-based outlier detection technique is "an anomaly is an observation which is suspected of being partially or wholly irrelevant because it is not generated by the stochastic model assumed" [Anscombe and Guttman 1960]. It is based on the key assumption: Normal data instances occur in high probability regions of a stochastic model, while anomalies occur in the low probability regions of the stochastic model. Statistical techniques fit a statistical model (usually for normal behavior) to the given data and then apply a statistical inference test to determine if an unseen instance belongs to this model or not. Instances that have a low probability from the applied test statistic are declared as outliers.

Pang et al. [2011, 2013] adapted the likelihood ratio test (LRT), which had previously been used mostly in epidemiological studies, to describe traffic patterns. They partitioned a city into uniform grids and counted the number of vehicles arriving in a grid over a time period. The objective was to identify contiguous sets of cells and time intervals that have the largest statistically significant departure from expected behavior (i.e., the number of vehicles). The regions whose log-likelihood ratio statistic value drops in the tail of χ^2 distribution are likely to be anomalous.

3.7.2. Disaster Detection and Evacuation. The Great East Japan Earthquake and the Fukushima nuclear accident caused large human population movements and evacuations. Understanding and predicting these movements are critical for planning effective humanitarian relief, disaster management, and long-term societal reconstruction. Lee and Sumiya [2010] aimed to detect events such as environmental disasters from geo-tagged tweet data. Song et al. [2013] constructed a large human mobility database that stores GPS records from mobile devices used by approximately 1.6 million people throughout Japan from August 1, 2010, to July 31, 2011. By mining this dataset, the short-term and long-term evacuation behaviors of individuals during this disaster were discovered. A probabilistic model was trained by the discovered evacuations and then applied to infer the population mobility in other cities impacted by the disasters. This research can inform decision making in future disaster relief and management.

4. TYPICAL TECHNOLOGY

In this section, we discuss five categories of techniques that are frequently used in urban computing: (1) urban sensing, (2) urban data management, (3) knowledge fusion across heterogeneous data, (4) dealing with data sparsity, and (5) urban data visualization.

4.1. Urban Sensing and Data Acquisition

The advances in sensing and data acquisition technologies have resulted in massive data in our cities, from traffic flow to air quality, from social media to geographic data. Here, we categorize the existing data acquisition technologies for urban computing into three folds: (1) traditional sensing and measurement, (2) passive crowd sensing, and (3) participatory sensing [Goldman et al. 2009], which will be detailed in later sections.

The first fold of technologies collects data through installing sensors dedicated to some applications, for example, burying loop sensors in roads for detecting traffic volume on road surfaces. The second fold of technologies leverages existing infrastructures to passively collect the data generated by crowds. For instance, wireless cellular networks are built for mobile communication between individuals. However, the mobile phone signal of a large number of people can be used to predict traffic conditions and improve urban planning. In the third fold, people actively obtain the information around them and contribute their own data to formulate collective knowledge that can solve a problem, in short, humans as sensors. Representative examples include detecting traffic congestion by aggregating the reports from a large number of people and probing the temperature throughout a city using the data shared by individual mobile phones. The major difference between the latter two folds of technologies is that people know what they are contributing and what the purpose for the sharing is in the third category (i.e., actively vs. passively). As the first fold of technology has already been widely used, we focus on introducing the latter two in this subsection.

4.1.1. Passive Crowd Sensing. To enable our modern lives, many advanced infrastructures have been built in cities (e.g., the ticketing systems of public transportation and wireless cellular networks). While these infrastructures were designed for other purposes, they can be used to sense city dynamics as well. The data produced by these infrastructures can also be analyzed for accomplishing other goals, such as to improve urban planning and ease traffic congestion.

- *Sensing City Dynamics with GPS-Equipped Vehicles*: Vehicles, such as taxis, buses, and private cars, have been equipped with a GPS sensor in recent years for different reasons (e.g., security, management, dispatch, and insurance measurement). The GPS sensors and communication modules enable these vehicles to report on their current location as well as other information to a backend center over a certain period. In fact, these GPS-equipped vehicles can be regarded as mobile sensors continually probing the traffic flow on road surfaces. The data acquired in these infrastructures also represents the city-wide human mobility patterns (e.g., if we know the pick-up and drop-off points of each taxi trip). A series of research has been done by acquiring the trajectory data of taxis, for example, a smart driving direction service [Yuan and Zheng et al. 2010a, 2011b, 2013b], sensing real-time gas consumption [Zhang et al. 2013], and detecting anomalies in a city [Liu et al. 2011; Chawla et al. 2012; Pan et al. 2013; Pang et al. 2011, 2013]. GPS-equipped buses also become prevalent in modern cities, majorly used for predicting the arrival time at bus stops. The GPS trajectory of these buses was also applied to traffic condition analysis [Bejan 2010] and bus route optimization [Bastani et al. 2011]. Some private vehicles are also embedded with a GPS sensor by insurance companies. The generated data, including GPS coordinates and other driving behaviors, is employed to measure the risk of car accidents that would happen to a driver, therefore determining an insurance package for the driver. The data can also be used to analyze the gas cost on different routes and educate drivers with eco-drive behaviors.

- *Data Acquisition Through Ticketing Systems of Public Transportation*: A variety of RFID-based cards have been used for charging people's commutes in public transportation systems, such as subways and buses. People usually need to swipe their card when entering a subway station or getting on a bus. Sometimes they also need to swipe at an exit or when getting off a bus. A transaction records consists of the money charged, timestamp, and location information, which may be a station, a dock, and a bus stop, or just the ID of a bus. If processed correctly, the origin and destination of each card holder can be inferred from the transaction records. As a result, we can model city-wide human mobility [Lathia and Capra 2011a, 2011b].
- *Data Acquisition Through Wireless Communication Systems*: Wireless communication systems (e.g., cellular networks and Wi-Fi), have been widely deployed in cities. The information that records the access of people to these wireless networks is actually another kind of footprint. For example, CDRs have been widely used in traffic and human mobility modeling [González et al. 2008; Candia et al. 2012]. A review of urban computing for mobile phone traces can be found in Jiang et al. [2013].
- *Data Acquisition Through Social Networking Services*: Advances in online social networking services have resulted in a large amount of social media, such as tweets, photos, and videos. Sometimes, social media is even associated with location information (e.g., a check-in at a venue or geo-tagged photos). The social media posted by users may describe the events that are happening around them, such as a natural disaster or a car accident. The real-time analysis of social media generated by massive users would help detect the anomalous events happening in a city. The geo-tagged social media may also reflect human mobility patterns in urban spaces, therefore enabling some useful applications, such as travel recommendations [Wei et al. 2012; Bao et al. 2012, 2014; Yoon et al. 2010, 2011] and location choosing for a business [Karamshuk et al. 2012].

4.1.2. Participatory Sensing. Thanks to the widespread adoption of powerful and networked (i.e., Internet-enabled) handheld devices, citizens are now taking a more active role in producing urban data. This trend has allowed a new category of applications to surface, in which information about our cities is collected by participants and is collectively used to offer services to citizens. We identify two main streams of work under the theme of participatory sensing: *human crowdsensing* and *human crowdsourcing*.

- *Human crowdsensing*: With this term, we refer to the case of users willingly contributing information as gathered from sensors embedded in the users' own devices. This can be, for example, GPS data from a user's mobile phone, as already explored in the Tiramisu project [Zimmerman et al. 2011], which is then used to estimate real-time bus arrivals. GPS data from users' personal devices is also exploited in traffic and navigation applications like Waze. In both cases, users simply need to start the application when taking a bus/car; without any further burden on their side, the application open on their phone passively contributes GPS data, which is then aggregated and analyzed for the application-specific goal (e.g., offering real-time bus arrival to other users, route computation). GPS data is only one example: users have been willingly contributing noise data, as picked up by the phone's microphone, along with the GPS location, to create urban noise maps [D'Hondt et al. 2011; Rana et al. 2010, 2013]. Sensing and mapping of environmental data, using personal sensing kits like SmartCitizen, are also gaining momentum: these devices can sense air quality, temperature, sound, humidity, light, CO_2, and NO_2. They are sufficiently cheap to be privately owned, thus paving the way for having hundreds or thousands of these devices spread around an urban area, potentially offering a very fine-grained spatiotemporal footprint on the liveability of our cities. As this human-collected data is intrinsically linked to where the people carrying the devices are, research is required to quantify bias in the data, to make explicit the extent to which the collected data is representative of actual environmental conditions [Mashhadi et al. 2013].

- *Human crowdsourcing*: With this term, we refer to scenarios where users are proactively engaged in the act of generating data, other than simply switching on/off an application or device. Examples include users generating reports on accidents, police traps, or any other road hazards to give other users in the area a "heads up" (Waze offers its users the ability to source this rich information on top of the sensed GPS data it already tracks from their mobile devices); cyclists annotating bike-friendly routes and reporting potholes and other types of problems that might affect fellow riders [Priedhorsky et al. 2010]; and citizens turning into cartographers to create open maps of their cities [Haklay and Weber 2008] or surveyors to report problems of local impact, so that councils can take action. The cognitive effort required in all these cases is much higher than with human crowdsensing, thus leading to open research questions in terms of users' motivation and long-term engagement that have only started to be explored [Hristova et al. 2013; Panciera et al. 2010].

4.2. Urban Data Management Techniques

The data generated in urban spaces is usually associated with a spatial or spatiotemporal property. For example, road networks and POIs are the frequently used spatial data in urban spaces; meteorological data, surveillance videos, and electricity consumption are temporal data (also called time series, or stream). Other data sources, like traffic flows and human mobility, have spatiotemporal properties simultaneously. Sometimes the temporal data can also be associated with a location, then becoming a kind of spatiotemporal data (e.g., the temperature of a region and the electricity consumption of a building). Consequently, good urban data management techniques should be able to deal with spatial and spatiotemporal data efficiently.

In addition, an urban computing system usually needs to harness a variety of heterogeneous data. In many cases, these systems are required to quickly answer users' instant queries (e.g., predicting traffic conditions and forecasting air pollution). Without the data management techniques that can organize multiple heterogeneous data sources, it becomes impossible for the following data-mining process to quickly learn knowledge from these data sources. For instance, without an efficient spatiotemporal indexing structure that well organizes POIs, road networks, traffic, and human mobility data in advance, the solely feature extraction process of the U-Air project [Zheng et al. 2013b] will last for a few hours. The delay will fail this application in telling people the air quality of a city every hour.

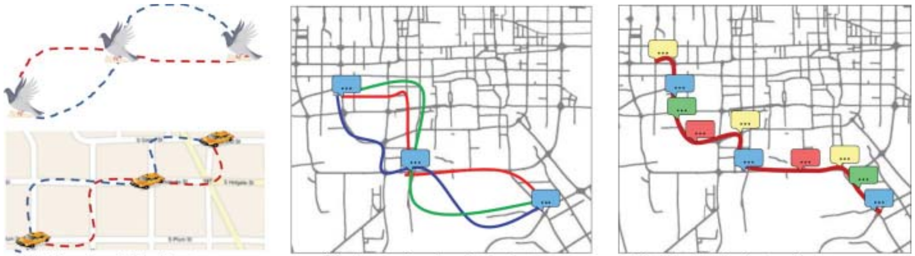
This section introduces three common data structures (i.e., stream, trajectory, and graph data) that are widely used in urban computing applications and the techniques for managing the three data structures. We also present some examples that integrate different data sources into a hybrid index.

4.2.1. Stream and Trajectory Data Management. *Stream data*, such as the temperature, electricity consumption, and video of surveillance cameras, is widely available in urban spaces. Managing and querying stream data have been studied intensively in the past decade [Aggarwal 2007; Lukasz 2010], with quite a few mature data stream management systems (DSMSs) like StreamInsight that have been built. A DSMS is a computer program to manage continuous data streams, similar to a database management system (DBMS), which is, however, designed for static data in conventional databases. In contrast to a DBMS, a DSMS executes a continuous query that produces new results as long as new data arrive at the system. Example queries include calculating the average temperature of buildings whose electricity consumption is higher than a threshold. One of the biggest challenges for a DSMS is to handle potentially infinite data streams using a fixed amount of memory and no random access to the data. There are two classes of approaches to limit the amount of data in one pass. One is compression techniques that try to summarize the data; the other is window techniques that try to portion the data into (finite) parts.

A *spatial trajectory* is a trace generated by a moving object in geographical spaces, usually represented by a series of chronologically ordered points, for example, $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, where each point consists of a geospatial coordinate set and a timestamp such as $p = (x, y, t)$. Many kinds of data generated in urban spaces can be formed as trajectories, from GPS traces of vehicles to a user's footprint (such as check-ins) in a location-based social network, from the call detail records of a mobile phone to the transaction records of a credit card. Though trajectory data can be regarded as a special case of stream data, the geographical position of each entry does make a difference and introduces many new problems calling for new techniques (refer to Zheng [2011f] for details):

1) *Data reduction techniques for trajectories*: Generally, the continuous movement of an object is recorded in an approximate form as discrete samples of location points. A high sampling rate of location points generates accurate trajectories but will result in a massive amount of data leading to enormous overhead in data storage, communications, and processing. Thus, it is vital to design data reduction techniques that compress the size of a trajectory while maintaining the utility of the trajectory. There are two major types of data reduction techniques running in a batch mode after the data is collected (e.g., Douglas-Peucker algorithm [Douglas and Peucker 1973]) or in an online mode as the data is being collected (such as the sliding window algorithm [Keogh et al. 2001; Maratnia 2004]). To evaluate a trajectory reduction technique, we usually consider the following three metrics: processing time, compression rate, and error measure (i.e., the deviation of an approximate trajectory from its original presentation). Recent research, PRESS [Song et al. 2014], has given the solution to the trajectory reduction on road networks. PRESS separates the spatial representation of a trajectory from the temporal representation, proposing a hybrid spatial compression algorithm and error-bounded temporal compression algorithm to compress the spatial and temporal information of a trajectory, respectively.

   In contrast to trajectory reduction algorithms that only focus on the spatiotemporal information of a trajectory, Chen et al. [2009] proposed to simplify a trajectory by considering both the shape skeleton and the semantic meanings of the trajectory. The algorithm is motivated by people's needs in a trajectory-sharing social networking site, like GeoLife [Zheng et al. 2008c and 2010d]. When browsing a trajectory



(a) Taxi and bird's traces (b) A user's check-in data (c) Check-in data of many users

Fig. 14. Reducing the uncertainty of a trajectory.

(probably standing for a travel route) shared by a user, the places where the user stayed, took photos, changed moving directions significantly, and so forth, would be more significant than other points in presenting the semantic meanings of the trajectory. Consequently, in this kind of trajectory-sharing system, GPS points with an important semantic meaning should be given a higher weight when choosing representative points for a simplified trajectory.

2) *Noise filtering techniques for trajectories*: A trajectory is usually generated with occasional outliers or some noisy points caused by the poor signal of location positioning systems. As a result, techniques for filtering the noisy points are needed for preprocessing spatial trajectories. General methods include mean and median filtering, the Kalman filter, and the particle filter. Refer to Chapter 1 of the book [Zheng and Zhou 2011f] for details.

3) *Techniques for indexing and query trajectories*: Querying the current location of a moving object has been studied extensively in moving object databases. The commonly used techniques include 3DR-Tree [Theodoridis et al. 1996] and MR-Tree [Xu et al. 1999]. Sometimes we need to search for historical trajectories satisfying certain criteria, for example, retrieving the trajectories of tourists passing a given region and within a time span (i.e., a spatiotemporal range query [Wang and Zheng et al. 2008]), taxi trajectories that pass a crossroad (i.e., a point query), or the trajectories that are similar to a query trajectory [Chen et al. 2010; Tang et al. 2011] (i.e., a trajectory query). See more techniques from Chapter 2 of Zheng [2011f].

4) *Techniques dealing with uncertainty of a trajectory*: Positioning devices are inherently imprecise, resulting in some uncertainty with regard to acquired locations of a moving object. Moreover, objects move continuously, while their locations can only be updated at discrete times. To save energy consumption and communication bandwidth, the time interval between two updates could exceed several minutes or hours, leaving the location of a moving object between two updates uncertain. For example, as shown in Figure 14(a), the time interval between two sampling points of a GPS-equipped taxi could be a few minutes, therefore having multiple possible paths traveling through the three sampled points.

   Map matching is to infer the path that a moving object like a vehicle has traversed on a road network based on the sampled trajectory. Map-matching techniques dealing with high-sampling-rate trajectories have already been commercialized in personal navigation devices, while those for low-sampling-rate trajectories [Lou et al. 2009] are still considered challenging. According to the result reported in Yuan et al. [2010b], given a trajectory with a sampling rate around 2 minutes per point, the highest accuracy of a map-matching algorithm is about 70%.

   When the time interval between consecutive sampling points becomes even longer (e.g., the interval between a user's two consecutive check-ins could be a few hours, and that for a bird's trace could be almost 1 day), existing map-matching algorithms do not work very well any longer. To address this issue, Wei et al. [2012] proposed to construct the most likely route passing a few sampled points based on many uncertain trajectories. For instance, as illustrated in Figure 14(c), if we put together the check-in data of many users, the uncertain route shown in Figure 14(b) could become certain that is, $uncertain + uncertain \rightarrow certain$.

Another branch of research is to predict a user's destination based on a partial trajectory [Krumm et al. 2006; Xue et al. 2013]. Here, a user's destination is uncertain at the very beginning and gradually becomes certain when a part of the trip has been traveled. A user's and other people's historical trajectories as well as other information, such as the land use of a location, can be used in destination prediction models.

5) *Trajectory pattern mining*: One branch of research is to find the *sequential patterns* from trajectories. Here, a sequential pattern means a certain number of moving objects traveling a common sequence of locations in similar travel time. Additionally, the locations in a travel sequence do not have to be consecutive. For instance, two trajectories A and B,

$$A : l_1 \overset{1.5h}{\rightarrow} l_2 \overset{1h}{\rightarrow} l_7 \overset{1.2h}{\rightarrow} l_4. \qquad B : l_1 \overset{1.2h}{\rightarrow} l_2 \overset{2h}{\rightarrow} l_4,$$

share a common sequence $l_1 \rightarrow l_2 \rightarrow l_4$, as the visiting orders and travel times are similar (though l_2 and l_4 are not consecutive in trajectory A). This is different from the longest common subsequence problem (LCSS) due to the temporal dimension. Giannotti et al. [2007] is the first research targeting at this problem. Xiao et al. [2010, 2012] proposed a graph-based sequence-matching algorithm to find the sequential pattern shared by two users' trajectories. The sequential patterns were then used to estimate the similarity between two users.

Another branch of research is to discover a group of objects that move together for a certain time period, such as flock [Gudmundsson et al. 2004, 2006], convoy [Jeung et al. 2008a, 2008b], swarm [Li et al. 2010], traveling companion [Tang et al. 2012, 2013], and gathering [Zheng et al. 2013b, 2014]. These concepts, which we refer to as group patterns, can be distinguished based on how the "group" is defined and whether they require the time periods to be consecutive. Specifically, a flock is a group of objects that travel together within a disc of some user-specified size for at least k consecutive timestamps. A major drawback is that a circular shape may not reflect the natural group in reality, which may result in the so-called lossy-flock problem [Jeung et al. 2008a]. To avoid rigid restrictions on the sizes and shapes of the group patterns, the convoy is proposed to capture a generic trajectory pattern of any shape and extent by employing the density-based clustering. Instead of using a disc, a convoy requires a group of objects to be density connected to each other during k consecutive time points. While both flock and convoy have strict requirements on consecutive time periods, Li et al. [2010] proposed a more general type of trajectory pattern, called swarm, which is a cluster of objects lasting for at least k (possibly nonconsecutive) timestamps. In contrast to flock, convoy, and swarm, which require a group pattern to contain the same set of individuals during its lifetime, gathering patterns allows members to enter and leave this group any time as long as a certain number of members can stay for a certain time period. This is more realistic as different people may join and leave an event frequently in a practical group event, such as a business promotion.

4.2.2. Graph Data Management. Graph is another kind of data format that is frequently used to represent urban data, such as road networks, subway systems, social networks, and sensor networks. Static graph data management [Angles and Gutierrez 2008] has been studied intensively in database areas for years with many mature management systems available. In urban computing, graphs are usually associated with a spatial property, resulting in many spatial graphs. For example, the node of a road network has a spatial coordinate, and each edge denoting a road segment has a spatial length. In many situations, these spatial graphs also contain temporal information. For instance, the traffic volume traversing a road segment changes over time, and the travel time between two landmarks is time dependent (e.g., Figure 6(a)). The structure of such a graph may also change over time. For instance, a traffic control may block the traffic flow between two locations, therefore temporarily removing the edge between the two locations. We call this kind of graph a *spatiotemporal graph* (ST graph) [Hong and Zheng et al. 2014]. Different from a *time-evolving graph,* which is usually used to represent a social network whose structures and properties also change over time, an ST graph has a spatial position for each node, resulting in a spatial distance between two nodes of the graph. The ST graph can be generated by projecting dynamic flow data onto a spatial graph (e.g., projecting the GPS trajectories of vehicles or call detail records of mobile phone users onto a road network). Other examples of spatiotemporal graphs are sensor networks, location-based social networks, and vehicle-to-vehicle networks, where the location of a node (e.g., a user or a vehicle) can change over time.

While managing spatiotemporal graphs effectively is very important to support the knowledge discovery process in urban computing, the corresponding data management techniques are somehow missing and yet to be explored (e.g., searching an ST graph for some subgraphs with total traffic volume above a threshold, or continuously finding the top-k clusters of spatially close nodes with relatively frequent communication among each other in an ST graph). This may be good news for researchers who are interested in graph data and spatial data management. Existing research majorly focuses on subgraph pattern mining [Zheng et al. 2011b] and time-dependent routing [Yuan et al. 2010a] on a spatiotemporal graph. For instance, the example (in Section 3.1.1) that gleans the problematic design in a city's road network according to human mobility data was formulated as a subgraph pattern-mining problem on a spatiotemporal graph. Recent research [Hong and Zheng et al. 2014] has started to detect from a spatiotemporal graph some black holes (or volcanos), which represent regions with the streaming-in traffic flow much larger than that streaming out. Sun et al. [2014] aimed to answer a spatial-temporal aggregate query in a location-based social network. Example queries include finding the top-k tourist attractions around a user that are most popular in the past 3 months.

4.2.3. Hybrid Indexing Structures. In an urban computing scenario, we usually need to harness a variety of data and integrate them into a data-mining model (see next section for details). This calls for hybrid indexing structures that can well organize different data sources.

For example, in many applications [Zheng et al. 2013b; Yuan et al. 2012a], we need to use POIs, road networks, traffic, and human mobility data simultaneously. Figure 15 presents a hybrid indexing structure, which combines a spatial index, hash tables, sorted lists, and an adjacency list. Specifically, a city is partitioned into grids by using a quad-tree-based spatial index. Each leaf node (grid) of the spatial index maintains two lists storing the POIs and road segments (only ID stored) that fall into the spatial range of the node. Further, each road segment ID points to two sorted lists. One is a list of taxi IDs sorted by their arrival time t_a at the road segment. Usually, we only need to store the ID of taxis traversing a road segment in the most recent hours. Consequently, a sorted list is fine here. When the time that needs to be stored is very long, a B-tree-based temporal index can be employed here to manage the ID of taxis. The other is a list of drop-off and pick-up points of passengers sorted by the pick-up time (t_p) and drop-off

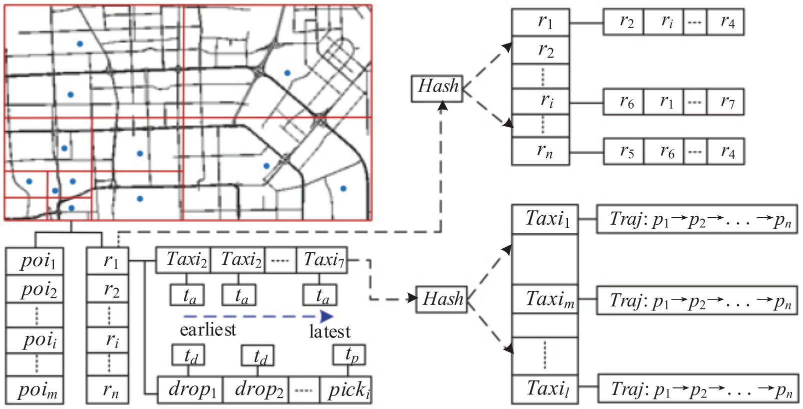


Fig. 15. A hybrid index managing road networks, POIs, and taxi trajectories.

time (t_d). The points of a taxi trajectory generated in recent hours can be maintained in a list that is stored in memory, while those of the historical trajectory are stored on a hard disk. Given the ID of a taxi, we can retrieve the trajectory of the taxi via a hash table. The structure of a road network is represented by an adjacency list shown on the top-right part of Figure 15. We can also use a hash table to retrieve the neighbor of a road segment in the road network it belongs to.

Using the air quality research [Zheng et al. 2013b] as an example, we describe how the hybrid indexing structure is used. The goal is to extract four categories of features (POI, road network, traffic, and human mobility features) for a given geographical region from different data sources belonging to the region. Given a geographical region, we first retrieve the leaf nodes that fall into the region from the quad-tree-based spatial index. By going through the lists of POIs and road networks in these leaf nodes respectively, we can quickly extract the POI features and road network features (e.g., the distribution of POIs across different categories and the total length of highways falling in the region). If we need to count the number of road intersections in a region, the adjacency list will be accessed via the hash table. As a road segment could cross a few regions, an optimal solution is to merge the road segment IDs retrieved from different leaf nodes before checking the adjacency list. The map matching [Yuan et al. 2010b] that projects a GPS trajectory onto a road network needs to access the quad-tree spatial index and the adjacency list simultaneously. After the map matching, we can update the taxi list and drop-off/pick-up list on a corresponding road segment. Later, the travel speed of each road segment can be calculated based on the trajectories of the taxis traversing the road segment. Similar to checking the road segment, we can merge the taxi IDs from different leaf nodes before accessing the trajectory lists. We can also calculate the number of people entering a region and that leaving a region based on the drop-off/pick-up list. The indexing structure introduced here is just an example and may not be the most optimal.

4.3. Knowledge Fusion across Heterogeneous Data Sources

In urban computing scenarios, we usually need to harness a variety of data sources, which calls for the technology that can effectively fuse the knowledge learned from multiple heterogeneous data sources. There are three major ways to achieve this goal: (1) Fuse different data sources at a feature level, that is, treat different data sources equally and put together the features extracted from different data sources into one feature vector. Of course, a certain kind of normalization technique should be applied to

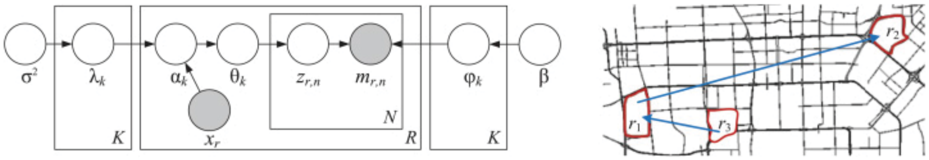


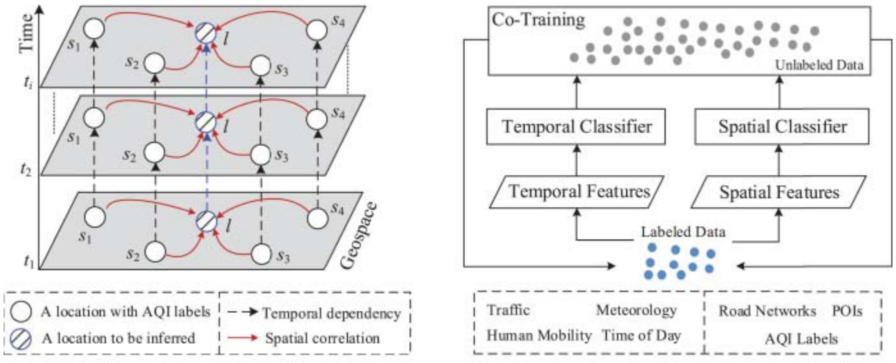
Fig. 16. Inferring functional regions in a city based on human mobility and POIs.

this feature vector before feeding it into a data analytics model. This is the most common way that we can see in data sciences dealing with heterogeneous sources. (2) Use different data at different stages. For instance, Zheng et al. [2011b] first partitioned a city into disjoint regions by major roads and then used human mobility data to glean the problematic configuration of a city's transportation network. This is also a very natural way when people think about data fusion. (3) Feed different datasets into different parts of a model simultaneously. This is based on the deep understanding of the data sources and algorithms. According to the studies [Zheng et al. 2013b; Yuan et al. 2012a], the third category of data fusion methods usually has a better performance beyond the first one, and the second category can be used simultaneously with the first and third. As the first and second categories are quite intuitive and have been extensively discussed in literature, we only focus on introducing the last one in this section with three tangible examples.

Example 1. The first example is what we have briefly introduced in Section 3.1.2, where Yuan et al. [2012a] inferred the functional regions in a city using road network data, points of interest, and human mobility learned from a large number of taxi trips. As depicted in Figure 16, an LDA-variant-based inference model was proposed, regarding a region as a document, a function as a topic, categories of POIs (e.g., restaurants and shopping malls) as metadata (like authors, affiliations, and key words), and human mobility patterns as words. More specifically, as shown in the right part of Figure 16, an individual departed from region r_1 at t_l and arrived at region r_2 at t_a, generating a commuting pattern $<r_1 \rightarrow r_2, t_l, t_a>$. Likewise, another three people generated another commuting pattern $<r_3 \rightarrow r_1, t_l', t_a'>$, respectively. The mobility pattern is defined as the commuting patterns between regions—when people leave a region and where they are heading to, and when people arrive at a region and where they came from. Each commuting pattern stands for one word describing a region, while the frequency of a commuting pattern denotes the number of occurrences of a word in a document. In this example, region r_1 contains two words $<r_1 \rightarrow r_2, t_l, t_a>$ and $<r_3 \rightarrow r_1, t_l', t_a'>$. The number of occurrences of the two words are one and three, respectively.

By feeding POIs (denoted as x_r) and human mobility patterns (denoted as $m_{r,n}$) into different parts of this model, a region is represented by a distribution of functions, each of which is further denoted by a distribution of mobility patterns. N stands for the number of words (i.e., mobility patterns in a region); R denotes the number of documents (regions); K is the number of topics, which should be predefined. Before running the model, a city was partitioned into disjointed regions using major roads such as highways and ring roads. So, this example uses the third category of data fusion techniques, combined with the second one.

Example 2. The second example was introduced in Section 3.3.2, which is about the inference of urban air quality using big data. As illustrated in Figure 17(a), air quality has the temporal dependency in an individual location (e.g., the AQI of a location tends to be good if the AQI of the past hour is also good) and the spatial correlation among different locations (e.g., the air quality of a place could be bad if the air qualities of its



(a) Philosophy of the inference model

(b) Procedure of co-training

Fig. 17. Inferring the fine-grained air quality throughout a city using big data.

surrounding locations are bad). A semisupervised learning approach was proposed to predict the air quality of a location without a monitoring station, based on a cotraining framework that consists of two separated classifiers. One is a spatial classifier based on an artificial neural network (ANN), which takes spatially related features (e.g., the density of POIs and length of highways) as input to model the spatial correlation between air qualities of different locations. The other is a temporal classifier based on a linear-chain conditional random field (CRF) involving temporally related features (e.g., traffic and meteorology) to model the temporal dependency of air quality in a location. The two classifiers are mutually reinforced in the cotraining framework. The results show its advantages beyond four categories of baselines, including linear/Gaussian interpolations, classical dispersion models, well-known classification models like decision tree and CRF, and ANNs.

The reason they fuse the data this way lies in three aspects: (1) The cotraining-based framework deals with data sparsity by leveraging unlabeled data. Though there is a huge volume of observation data, such as traffic flow, the labeled data generated by existing monitoring stations is very limited (i.e., a label sparsity problem). (2) There happen to be two sets of features providing two different views for an instance (i.e., the air quality of a location). If simply putting together the spatially related features (such as the structure of road networks) with the temporally related features (such as meteorology and traffic flow) whose values change over time constantly, the spatially related features will be ignored by some machine-learning models. That is, no matter what the air quality is in a location, these spatially related features do not change over time at all. (3) The two classifiers model the spatial correlation and temporal dependency respectively, which is interpretable.

Example 3. This example was mentioned in Section 3.4 as an application about energy consumption. Specifically, a taxi's refueling event was detected from its GPS trajectories, using first a spatiotemporal clustering algorithm to identify the locations where the taxi stayed for a while and then a classification algorithm to filter some instances that may not be real refueling events, such as waiting for a traffic light close to a gas station. If the queuing time of a gas station can be detected from the data, the number of vehicles in the queue can be calculated according to the classic queue theory. Consequently, the gas consumption can be roughly estimated, supposing the volume of gas with which each vehicle is refueled follows a certain distribution. However, at some moment, many gas stations may not have a taxi waiting in a queue (but there would be other vehicles), leading to a data-missing problem. In addition, the distribution of

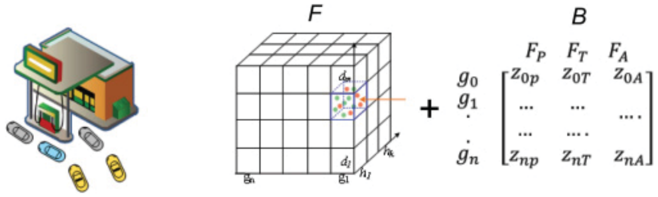


Fig. 18. Sensing urban refueling behavior with GPS-equipped taxis.

taxis in gas stations may be skewed from that of normal vehicles. Observing more taxis (or fewer taxis) in a station does not suggest more normal vehicles (or fewer normal vehicles).

To address this issue, as shown in Figure 18, a tensor F was formulated with the three dimensions denoting gas stations, day of the week, and time of day, respectively. Given that the tensor was very sparse, a tensor decomposition technology was applied to approximate the tensor with the multiplication of three low-rank matrices and a core tensor. In order to achieve a better approximation, a feature matrix B was also built based on three other data sources, consisting of the POIs and traffic flow around a station as well as the geospatial size of the station, where each row stands for a gas station and each column denotes a feature. The general idea is that gas stations with similar features (such as surrounding traffic patterns, POI distributions, and the structure of road networks) could have similar refueling patterns. As the feature matrix is quite dense (not sparse), we can enhance the accuracy of estimating the missing values in the tensor by incorporating the matrix into the process of tensor decomposition. Note that the number of taxis was not used to infer the number of normal vehicles. Instead, the waiting time of taxis, which is transferrable, was employed to estimate the length of the queue.

This is a clear example of computing with heterogeneous data sources, consisting of POIs, road networks, layout of gas stations, and GPS trajectories of taxicabs, where the first three data sources were fed into a feature matrix and the last one was used to formulate a tensor. The matrix and tensor were blended in the tensor decomposition-based collaborative filtering model (refer to Section 4.4.3 for more details on the decomposition technique).

4.4. Techniques Dealing with Data Sparsity

There are many reasons that lead to a data-missing problem. For example, a user would only check in at a few venues in a location-based social networking service, and some venues may not have people visiting them at all. If we put user–location into a matrix where each entry denotes the number of visits of users to a place, the matrix is very sparse; that is, many entries do not have a value. If we further consider the activities (such as shopping, dining, and sports) that a user can perform in a location as the third dimension, a tensor can be formulated. Of course, the tensor is even sparser. Similarly, in the application mentioned in Section 4.3, Example 3, many gas stations do not really have a taxi waiting in a queue at some moments. Consequently, the tensor shown in Figure 18 is also sparse. The application presented in Section 4.3, Example 2 also has a data sparsity problem as there are only a few air quality monitoring stations generating training data but having thousands of places in a city to infer.

Data sparsity is a general challenge that has been studied for years in many computing tasks. Instead of proposing new algorithms, we hereafter discuss three categories of techniques (but not limited to the three) that can be applied to tackle the data sparsity problems in urban computing:

$$M = \begin{array}{c} \\ u_1 \\ u_2 \\ .. \\ u_n \end{array} \begin{array}{c} \begin{array}{ccccc} l_1 & l_2 & l_3 & \ldots & l_k \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & . & 0 \\ 1 & 1 & 2 & . & 0 \\ 0 & 0 & 1 & . & 0 \\ 0 & 0 & 0 & . & 1 \end{bmatrix} \end{array}$$

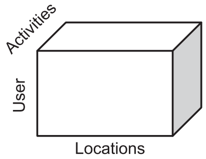


Fig. 19. An example of matrix and tensor.

4.4.1. Collaborative Filtering. Collaborative filtering (CF) is a well-known model widely used in recommender systems. The general idea behind collaborative filtering is that similar users make ratings in a similar manner for similar items [Goldberg et al. 1992; Nakamura et al. 1998]. Thus, if similarity is determined between users and items, a potential prediction can be made as to the rating of a user with regard to future items. Depending on the applications of urban computing, items can be POIs, such as restaurants and gas stations; road segments; geographical regions; and so forth, and users can be drivers, passengers, or subscribers of a service. Once formulating a matrix, we can use a CF model to fill the missing values.

Memory-based CF is the most widely used algorithm, which consists essentially of heuristics that make rating predictions based on the entire collection of previously rated items by users. That is, the value of the unknown rating for a user and an item is usually computed as an aggregate of the ratings of some other (usually, the N most similar) users for the same item. There are two classes of memory-based CF models: user-based and item-based techniques. Using the user–location matrix shown in Figure 19 as an example, user p's interest (r_{pi}) in a location i can be predicted according to the following three equations, which is a common implementation of user-based collaborative filtering:

$$r_{pi} = \overline{R_p} + d \sum_{u_q \in U'} sim(u_p, u_q) \times (r_{qi} - \overline{R_q}) \tag{1}$$

$$d = \frac{1}{|U'|} \sum_{u_q \in U'} sim(u_p, u_q) \tag{2}$$

$$\overline{R_p} = \frac{1}{|S\left(R_p\right)|} \sum_{i \in S(R_p)} r_{pi}, \tag{3}$$

where $sim(u_p, u_q)$ denotes the similarity between user u_p and u_q; $\overline{R_q}$ and $\overline{R_p}$ mean the average rating of u_p and u_q, respectively; $S(R_p)$ represents the collection of locations visited by u_p; and U' is the collection of users who are the most similar to u_q. There are other implementation methods of CF models. Refer to Nakamura et al. [1998] for details.

4.4.2. Matrix Factorization. Matrix factorization decomposes a matrix into a production of two or three matrices. There are multiple kinds of matrix factorizations, for example, LU decomposition, QR decomposition, and SVD (singular value decomposition). SVD is one of the most frequently used matrix decomposition methods in collaborative filtering, factorizing a matrix X into three matrices, consisting of left singular vectors (U), singular values (Σ), and right singular vectors, as shown in Figure 20. When the matrix X is very sparse, we usually can approximate it with three low-rank matrices. For instance, we can choose the top n biggest singular values (Σ) whose summation is larger than 90% of total summation of all the singular values. In this way, matrix factorization can be used as an efficient method to implement collaborative filtering.

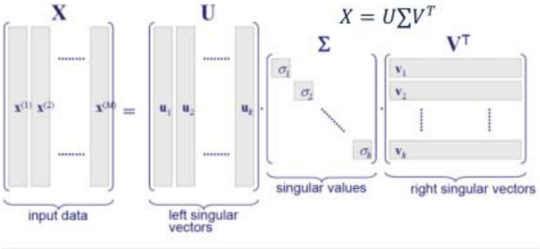


Fig. 20. SVD-based matrix factorization.

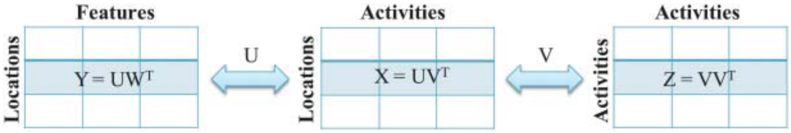


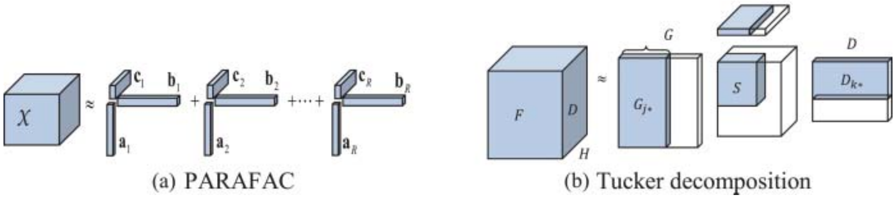
Fig. 21. Coupled matrix factorization for location–activity recommendation.

Sometimes we can also consider contexts during the process of matrix factorization. For example, as shown in Figure 21, given many users' location histories, Zheng et al. [2010f] built a location–activity matrix (X), where rows stand for locations (such as restaurants and shopping malls) and columns represent activities. An entry in the matrix denotes the frequency of an activity that has been performed by people in a particular location. If this location–activity matrix is completely filled, we can recommend a set of locations for a particular activity by retrieving the top-k locations with a relatively high frequency from the column that corresponds to that activity. Likewise, when performing activity recommendations for a location, the top-k activities can be retrieved from the row corresponding to the location. However, the location–activity matrix is incomplete and very sparse, as an individual can usually visit very few locations. Accordingly, a traditional CF model does not work very well in generating quality recommendations. Solely factorizing X does not help much either as the data is oversparse.

To address this issue, the information from another two matrices, respectively shown in the left and right part of Figure 21, can be incorporated into the matrix factorization. One is a location–feature matrix; the other is an activity–activity matrix. Such kinds of additional matrices are usually called contexts, which can be learned from other data sources. In this example, Matrix Y, where a row stands for a location and a column denotes a category of POIs (such as restaurants and hotels) that fall in the location, can be built from a POI database. Matrix Z models the correlation between two different activities, which can be learned from the search results by sending the titles of two activities into a search engine. The main idea is to propagate the information among X, Y, and Z by requiring them to share low-rank matrices U and V in a collective matrix factorization model. More specifically, an objective function was formulated as:

$$\begin{aligned} L(U, V, W) = \; & \frac{1}{2} \parallel I \circ (X - UV^T) \parallel_F^2 + \frac{\lambda_1}{2} \parallel Y - UW^T \parallel_F^2 \\ & + \frac{\lambda_2}{2} \parallel Z - VV^T \parallel_F^2 + \frac{\lambda_3}{2} \left(\parallel U \parallel_F^2 + \parallel V \parallel_F^2 + \parallel W \parallel_F^2 \right), \end{aligned}$$

where $\parallel \cdot \parallel_F$ denotes the Frobenius norm; I is an indicator matrix with its entry $I_{ij} = 0$ if X_{ij} missing, $I_{ij} = 1$ otherwise; And the operator $\circ$ denotes the entry-wise product. The first three terms in the objective function control the loss in matrix factorization, and



(a) PARAFAC

(b) Tucker decomposition

Fig. 22. Tensor decomposition.

the last term controls the regularization over the factorized matrices so as to prevent overfitting. In general, this objective function is not jointly convex to all the variables U, V, and W. Consequently, some numerical method, such as gradient descent, was used to get local optimal solutions.

4.4.3. Tensor Decomposition. A tensor, which usually has three dimensions, can be decomposed into the multiplication of matrices or vectors based on the entries with values. The objective function guiding the decomposition is to minimize the error between the multiplication of the decomposed results and the values of the existing entries in the tensor. After the decomposition, we can fill the missing values in a tensor through multiplying the decomposed matrices or vectors. Frequently used tensor decomposition methods include PARAFAC [Rro 1997] and Tucker decomposition [Kolda and Bader 2009]. As illustrated in Figure 22(a), PARAFAC decomposes a tensor into the summation of a series of multiplication of three vectors, while Tucker decomposition approximates a tensor with the multiplication of three matrices and a core tensor, as depicted in Figure 22(b). Sometimes, given a certain approximation error, we can only maintain the first few rows or columns (all called singular vectors) of a decomposed matrix to achieve a better efficiency (especially when the tensor is very sparse), formally denoted as:

$$F_{ijk} = S \times_H H \times_G G \times_D D \approx S \times_H H_{i*} \times_G G_{j*} \times_D D_{k*}$$

where H_{i*}, G_{j*}, and D_{k*} denote low-rank representation of matrices H, G, and D, respectively (see Figure 22(b)), and $\times_H$ means tensor matrix multiplication according to dimension H (also called mode of a tensor). Refer to a tutorial on tensor decomposition [Faloutsos et al. 2007] for details.

To enhance the accuracy of decomposing a sparse tensor, context information can be added into the decomposition process. This is similar to matrix factorization with contexts. Following the location–activity recommendation example shown in Figure 21 (Section 4.4.2), Zheng et al. [2010a, 2012d] further took a user dimension into account in a recommendation system, therefore generating a (user–location–activity) tensor. Intrinsically, the tensor is very sparse as a user usually visits a few places. Later, as demonstrated in Figure 23, four matrices were formulated based on other data sources, used as the context information to improve the accuracy of the tensor decomposition. A PARAFAC-style tensor decomposition framework was then proposed to incorporate the tensor with these context matrices for a regularized decomposition. More specifically, an objective function was defined as follows:

$$\begin{aligned}\mathcal{L}(X, Y, Z, U) \;=\; & \frac{1}{2}||\mathcal{A} - [\![X, Y, Z]\!]||_F^2 + \frac{\lambda_1}{2}\mathrm{tr}\big(X^T L_B X\big) + \frac{\lambda_2}{2}||C - YU^T||_F^2 \\ & + \frac{\lambda_3}{2}\mathrm{tr}\big(Z^T L_D Z\big) + \frac{\lambda_4}{2}||E - XY^T||_F^2 \\ & + \frac{\lambda_5}{2}\big(||X||_F^2 + ||Y||_F^2 + ||Z||_F^2 + ||U||_F^2\big),\end{aligned}$$

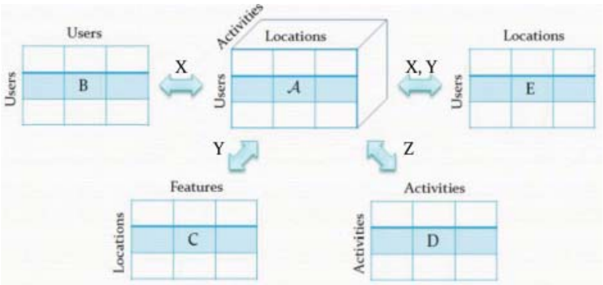


Fig. 23. Tensor-decomposition-based location–activity recommendation.

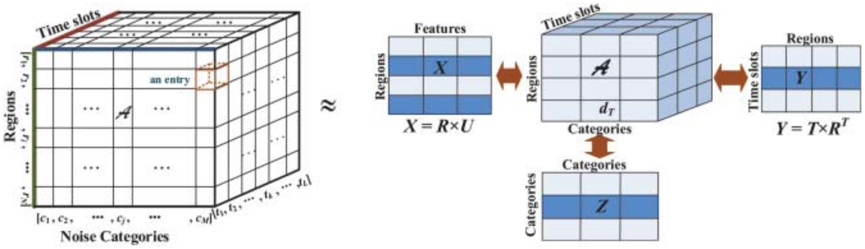


Fig. 24. Using context-aware tensor decomposition to infer urban noise.

where $[\![X, Y, Z]\!] = \sum \boldsymbol{x}_i \circ \boldsymbol{y}_i \circ \boldsymbol{z}_i$; the operator $\circ$ denotes the outer product; L_B is the Laplacian matrix of B, defined as $L_B = Q - B$, with Q being a diagonal matrix whose diagonal entries $Q_{ij} = \sum_j B_{ij}$; $\mathrm{tr}(\cdot)$ denotes the trace of a matrix; $\|\cdot\|_F$ denotes the Forbenius norm; and $\lambda_i (i = 1, \ldots, 5)$ are tunable model parameters. Given the objective function, a gradient descent was employed to find a local minimal result for X, Y, and Z.

The similar tensor decomposition technology was also employed by Zheng et al. [2014b] to infer urban noises. As shown in the left part of Figure 24, the noises in each geographical region are modeled by a tensor, $\mathcal{A} \in \mathbb{R}^{N \times M \times L}$, with three dimensions denoting N regions, M noise categories, and L time slots, respectively. A common approach to supplement the missing entries of tensor $\mathcal{A}$ is to decompose $\mathcal{A}$ into the multiplication of a core tensor $S \in \mathbb{R}^{d_R \times d_C \times d_T}$ and three matrices, $R \in \mathbb{R}^{N \times d_R}$, $C \in \mathbb{R}^{M \times d_C}$, and $T \in \mathbb{R}^{L \times d_T}$, using a Tucker decomposition model [Kolda and Bader 2009]. The objective function to control the error of the decomposition is usually defined as:

$$\mathcal{L}(S, R, C, T) = \frac{1}{2} ||\mathcal{A} - S \times_R R \times_C C \times_T T||_F^2 + \frac{\lambda}{2} \left(||R||_F^2 + ||C||_F^2 + ||T||_F^2 \right)$$

In this problem, however, the tensor is oversparse. For example, if setting 1 hour as a time slot, only 5.18% of the entries of $\mathcal{A}$ have values in weekends. Decomposing $\mathcal{A}$ solely based on its own nonzero entries is not accurate enough. To deal with the data sparsity problem, as illustrated in the right part of Figure 24, Zheng et al. extracted three categories of features, consisting of geographical features, human mobility features, and the noise category correlation features (denoted by matrices X, Y, and Z), from POI/road network data, user check-ins, and 311 data, respectively. These features were used as contexts in the decomposition process to reduce inference errors, using the objective function:

$$\mathcal{L}(S,R,C,T,U) = \frac{1}{2}||\mathcal{A} - S \times_R R \times_C C \times_T T||_F^2 + \frac{\lambda_1}{2}||X - RU||_F^2 + \frac{\lambda_2}{2}\mathrm{tr}(C^T L_Z C) + \frac{\lambda_3}{2}||Y - TR^T||_F^2 + \frac{\lambda_4}{2}(||R||_F^2 + ||C||_F^2 + ||T||_F^2 + ||U||_F^2),$$

where $||\mathcal{A} - S \times_R R \times_C C \times_T T||_F^2$ is to control the error of decomposing $\mathcal{A}$; $||X - RU||_F^2$ is to control the error of factorization of X; $||Y - TR^T||_F^2$ is to control the error of factorization of Y; $||R||_F^2 + ||C||_F^2 + ||T||_F^2 + ||U||_F^2$ is a regularization penalty to avoid overfitting; and λ_1, λ_2, λ_3, and λ_4 are parameters controlling the contribution of each part during the collaborative decomposition. When $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 0$, our model degenerates to the original Tucker decomposition. $C \in \mathbb{R}^{M \times d_C}$, $tr(\cdot)$ denotes the matrix trace; $D_{ii} = \sum_i Z_{ij}$ is a diagonal matrix; and $L_Z = D - Z$ is the Laplacian matrix of the category correlation graph.

More specifically, $\mathcal{A}$ and X share matrix R; $\mathcal{A}$ and Y share matrix R and T; and L_Z influences factor matrix C. The dense representation of X, Y, and Z contributes to the generation of a relatively accurate R, C, and T, which reduce the decomposition error of $\mathcal{A}$ in turn. In other words, the knowledge from geographical features, human mobility features, and the correlation between noise categories are propagated into tensor $\mathcal{A}$.

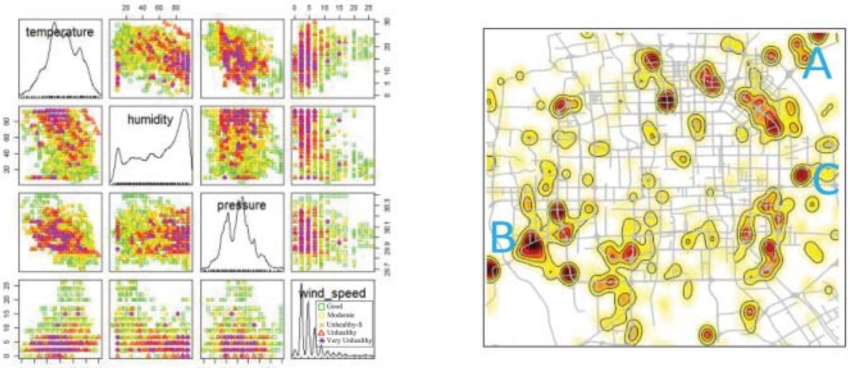
4.4.4. Semisupervised Learning and Transfer Learning. *Semisupervised learning* is a class of supervised learning tasks and techniques that also make use of unlabeled data for training—typically a small amount of labeled data with a large amount of unlabeled data. Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. There are multiple semisupervised learning methods, such as generative models, graph-based methods, and cotraining. A survey on this topic can be found in Zhu [2008]. Specifically, cotraining is a semisupervised learning technique that requires two *views* of the data. It assumes that each example is described by two different feature sets that provide different and complementary information about an instance. Ideally, the two feature sets of each instance are conditionally independent given the class, and the class of an instance can be accurately predicted from each view alone. Cotraining can generate a better inference result as one of the classifiers correctly labels data that the other classifier previously misclassified [Nigam and Ghani 2000]. Example 2 introduced in Section 4.3.2 is based on the cotraining technique.

Transfer learning: A major assumption in many machine-learning and data-mining algorithms is that the training and future data must be in the same feature space and have the same distribution. However, in many real-world applications, this assumption may not hold. For example, we sometimes have a classification task in one domain of interest, but we only have sufficient training data in another domain of interest, where the latter data may be in a different feature space or follow a different data distribution. Different from semisupervised learning, which assumes that the distributions of the labeled and unlabeled data are the same, transfer learning, in contrast, allows the domains, tasks, and distributions used in training and testing to be different. In the real world, we observe many examples of transfer learning. For instance, learning to recognize tables may help recognize chairs. Pan et al. [2010] gave a good survey on transfer learning, classifying transfer learning into three subcategories based on different situations between the source and target domains and tasks, as shown in Table I.

Transfer learning algorithms can help deal with data sparsity problems in urban computing. For instance, an alternative way to conquer the labeling sparsity problem

Table I. Different Kinds of Transfer Learning

Learning settings		Source and target domains	Source and target tasks
Traditional machine learning		Same	Same
Transfer learning	Inductive learning/unsupervised transfer learning	Same	Different but related
		Different but related	Different but related
	Transductive learning	Different but related	Same



(a) Correlation between meteorology and air quality (b) Refueling behavior inferred from taxi data

Fig. 25. Examples of visualization on urban data.

in the air quality inference [Zheng 2013b] is to transfer the knowledge learned from some cities with sufficient air quality data to the cities having insufficient data. This belongs to transductive learning as shown in Table I.

4.5. Visualizing Big Urban Data

When talking about data visualization, many people would only think about (1) the visualization of raw data and (2) the presentation of results generated by data-mining processes [Martinoc 2007]. The former may reveal the correlation between different factors, therefore suggesting features for a machine-learning model. For instance, Figure 25(a) shows the correlation matrix between the AQI of PM_{10} and meteorological data, consisting of temperature, humidity, barometer pressure, and wind speed, using the data collected in Beijing from August to December 2012, where each row/column denotes one kind of meteorological data and a plot means the AQI label of a location. Apparently, a high wind speed disperses the concentration of PM_{10}, and high humidity usually causes a high concentration. Consequently, they can be important features in a machine-learning model to infer the air quality of a location.

On the other hand, Figure 25(b) visualizes the result (i.e., the number of visits to gas stations by all the drivers in a city) inferred by the data-mining model we introduced in Example 3 of Section 4.3. The presentation of results can help energy infrastructure authorities better make a decision on where additional gas stations should be built. As mentioned before, spatiotemporal data is widely used in urban computing. For a comprehensive analysis, the data needs to be considered from two complementary perspectives: (1) as spatial distributions changing over time (i.e., spaces in time) and (2) as profiles of local temporal variation distributed over space (i.e., time in spaces) [Andrienko 2010].

However, data visualization is not solely about displaying raw data and presenting results. Exploratory visualization becomes even more important in urban computing, detecting and describing patterns, trends, and relations in data, motivated by certain purposes of investigation. As something relevant is detected in data, new questions arise, causing specific parts to be viewed in more detail. So, exploratory visualization combines the strengths of human and electronic data processing in an interactive way, involving hypothesis generation rather than mere hypothesis testing [Andrienko 2003].

4.6. Other Techniques

Besides the aforementioned techniques, urban computing, as a multidisciplinary field, also needs the support of other technologies, such as optimization technology and information security. Though trying to involve the knowledge of other fields as much as possible, this article is majorly written from the computer sciences' perspective.

4.6.1. Optimization Techniques. First, many data-mining tasks can be solved by optimization methods, such as matrix factorization and tensor decomposition. Examples include the location–activity recommendations [Zheng et al. 2010a, 2010f, 2012d] and the refueling behavior inference research [Zhang et al. 2013] we introduced in Section 4.3. Second, the learning process of many machine-learning models is actually based on optimization and approximation algorithms, for example, maximum likelihood, gradient descent, and EM (estimation and maximization). Third, the research results from operation research can be applied to solving an urban computing task if combined with other techniques, such as database algorithms. For instance, the ridesharing problem has been studied for many years in operation research. It has been proved to be an NP-hard problem if we want to minimize the total travel distance of a group of people who expect to share rides. As a consequence, it is really hard to apply existing solutions to a large group of users, especially in an online application. In the dynamic taxi ridesharing system T-Share, Ma et al. [2013] combined spatiotemporal database techniques with optimization algorithms to significantly scale down the number of taxis that needs to be checked. Finally, the service can be provided online to answer the instant queries of millions of users. Another example was introduced in Section 3.7.1. Chawla et al. [2012] combined a PCA-based anomaly detection algorithm with L_1 minimization techniques to diagnose the traffic flows that lead to a traffic anomaly. The spatiotemporal property and dynamics of urban computing applications also bring new challenges to current operation research.

4.6.2. Information Security. Information security is also nontrivial for an urban computing system that may collect data from different sources and communicate with millions of devices and users. The common problems that would occur in urban computing systems include data security (e.g., guaranteeing the received data is integrated, fresh, and undeniable), authentication between different sources and clients, and intrusion detection in a hybrid system (connecting digital and physical worlds).

4.7. Future Directions

Although many research projects about urban computing have been done in recent years, there are still quite a few technologies that are missing or not well studied.

- *Balanced crowdsensing*: The data generated through a crowdsensing method is nonuniformly distributed in geographical and temporal spaces. In some locations, we may have much more data than what we really need. A down-sampling method (e.g., compressive sensing) could be useful to reduce a system's communication loads. On the contrary, in the places where we may not have enough data or even do not have data at all, some incentives that can motivate users to contribute data should be considered. Given a limited budget, how to configure the incentive for different locations and time periods so as to maximize the quality of the received data (e.g., the coverage or accuracy) for a specific application has yet to be explored.

- *Skewed data distribution*: In many cases, what we can obtain is a sample of the urban data, whose distribution may be skewed from the complete dataset. Having the entire dataset may be always infeasible in an urban computing system. Some information is transferrable from the partial data to the entire dataset. For instance, the travel speed of taxis on roads can be transferred to other vehicles that are also traveling on the same road segment. Likewise, the waiting time of a taxi at a gas station can be used to infer the queuing time of other vehicles. Other information, however, cannot be directly transferred. For example, the traffic volume of taxis on a road may be different from private vehicles. As a consequence, observing more taxis on a road segment does not always suggest more other vehicles.
- *Managing and indexing multimode data sources*: Different kinds of index structures have been proposed to manage different types of data individually, whereas the hybrid index that can simultaneously manage multiple types of data (e.g., spatial, temporal, and social media) is yet to be studied. The hybrid index, such as the example shown in Figure 15, is a foundation enabling an efficient and effective learning of multiple heterogeneous data sources.
- *Knowledge fusion*: Data-mining and machine-learning models dealing with a single data source have been well explored. However, the methodology that can learn mutually reinforced knowledge from multiple data sources is still missing. The fusion of knowledge does not mean simply putting together a collection of features extracted from different sources but also requires a deep understanding of each data source and an effective usage of different data sources in different parts of a computing framework. The research, like the three examples [Zheng et al. 2013b; Zhang et al. 2013] presented in Section 4.3, is considered rare.
- *Exploratory and interactive visualization for multiple data sources*: An urban computing system usually has a lot of data and knowledge to visualize. So far, it is not an easy task to investigate the implicit relationship among multiple data sources through an exploratory visualization in spatial and spatiotemporal spaces. For instance, there are multiple factors (e.g., traffic, factory emission, meteorology, and land use) that could influence the air quality of a location. Unfortunately, it is still not easy to answer the following questions: Which factor is more prominent in impacting the air quality of a given location or in a given time period? What is the major root cause of $PM_{2.5}$ in the winter of Beijing?
- *Algorithm integration*: To provide an end-to-end urban computing scenario, we need to integrate algorithms of different domains into a computing framework. For instance, we need to combine data management techniques with machine-learning algorithms to provide both an efficient and effective knowledge discovery ability. Similarly, through integrating spatiotemporal data management algorithms with optimization methods, we can solve the large-scale dynamic ridesharing problem. Visualization techniques should be involved in a knowledge discovery process, working with machine-learning and data-mining algorithms. So, urban computing calls for both the fusion of data and the integration of algorithms. In the long run, the unprecedented data that we are facing will blur the boundary between different domains in conventional computer sciences (e.g., databases and machine learning) or even bridge the gap between different disciplines' theories, such as civil engineering and ecology.
- *Intervention-based analysis and prediction*: In urban computing, it is vital to predict the impact of a change in a city's setting. For instance, how will a region's traffic change if a new road is built in the region? To what extent will air pollution be reduced if we remove a factory from a city? How will people's travel patterns be

affected if a new subway line is launched? Being able to answer these kinds of questions with automated and unobtrusive technologies will be tremendously helpful to inform governmental officials' and city planners' decision making. Unfortunately, the intervention-based analysis and prediction technology that can estimate the impact of a change in advance by plugging in and out some factors in a computing framework is not well studied yet.

4.8. Miscellaneous

Conferences and workshops: The research into urban computing published in the computer sciences domain can be majorly found in leading conferences, such as KDD, ICDE, and UbiComp, and a few workshops, like the ACM International Workshop on Urban Computing (UrbComp) [Zheng and Wolfson 2012c; Zheng et al. 2013a].

Journals and magazines: We can also easily find related articles from many journals, such as *IEEE Transaction on Knowledge Discovery and Data Engineering* (IEEE TKDE), *ACM Transaction on Intelligent Systems and Technology* (ACM TIST), and *Personal and Ubiquitous Computing* (Springer PUC), and magazines, such as *IEEE Pervasive Computing*.

Data sources: Quite a few major cities have their own open data portal; for example, New York City has published many useful data sources on their portal: https://nycopendata.socrata.com/. There are also a few public datasets on some researchers' homepages, for example, Dr. Zheng's homepage http://research.microsoft.com/en-us/people/yuzheng/.

5. CONCLUSION

The massive amount of data that has been generated in urban spaces and the advances in computing technology have provided us with unprecedented opportunities to tackle the big challenges that cities face. Urban computing is an interdisciplinary field where computer sciences meet conventional city-related disciplines, such as civil engineering, ecology, sociology, economy, and energy. In the context of cities, the vision of urban computing—acquisition, integration, and analysis of big data to improve urban systems and life quality—will lead to smarter and greener cities that are of great importance to billions of people. The big data will also blur the boundary between different domains that were formulated in conventional computer sciences (e.g., databases, machine learning, and visualization) or even bridge the gap between different disciplines (e.g., computer sciences and civil engineering). While urban computing holds great promise to revolutionize urban sciences and progress, quite a few techniques, such as the hybrid indexing structure for multimode data, the knowledge fusion across heterogeneous data sources, exploratory visualization for urban data, the integration of algorithms of different domains, and intervention-based analysis, are yet to be explored. This article discussed the concept, framework, and challenges of urban computing; introduced the representative applications and techniques for urban computing; and suggested a few research directions that call for efforts from the communities.

REFERENCES

C. Aggarwal. 2007. *Data Streams: Models and Algorithms*. Springer, New York.

G. Andrienko, N. Andrienko, S. Bremm, T. Schreck, T. V. Landesberger, P. Bak, and D. Keim. 2010. Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns. In *Proceedings of the Eurographics/IEEE-VGTC Symposium on Visualization 2010*.

N. Andrienko, G. Andrienko, and P. Gatalsky. 2003. Exploratory spatio-temporal visualization: An analytical review. *Journal of Visual Languages and Computing* 14, 6, 503–541.

R. Angles and C. Gutierrez. 2008. Survey of graph database models. *ACM Computing Surveys* 40, 1, 1–39.

F. J. Anscombe and I. Guttman. 1960. Rejection of outliers. *Technometrics* 2, 2, 123–147.

J. Bao, Y. Zheng, D. Wilkie, and M. F. Mokbel. 2012. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th ACM SIGSPATIAL Conference on Advances in Geographic Information Systems*. ACM, 199–208.

J. Bao, Y. Zheng, D. Wilkie, and M. F. Mokbel. 2014. A survey on recommendations in location-based social networks. Submitted to GeoInformatica, 2015.

F. Bastani, Y. Huang, X. Xie, and J. W. Powell. 2011. A greener transportation mode: flexible routes discovery from GPS trajectory data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 405–408

A. I. Bejan, R. J. Gibbens, D. Evans, A. R. Beresford, J. Bacon, and A. Friday. 2010. Statistical modelling and analysis of sparse bus probe data in urban areas. In *Proceedings of the 13th IEEE International Conference on Intelligent Transportation Systems*. IEEE, 1256–1263.

M. Berlingerio Calabrese, F. Giusy, D. Lorenzo, R. Nair, F. Pinelli, and M. L. Sbodio. 2013. AllAboard: A system for exploring urban mobility and optimizing public transport using cellphone data. In *Proceedings of the 12th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases.* Spring Press, 663–666.

P. Borgnat, E. Fleury, C. Robardet, and A. Scherrer. 2009. Spatial analysis of dynamic movements of VloV, Lyon's shared bicycle program. In *Proceedings of the European Conference on Complex Systems*. Warwick University, Coventry, UK.

J. Candia, M. C. González, P. Wang, T. Schoenharl, G. Madey, and A. L. Barabási. 2012. Uncovering individual and collective human dynamics from mobile phone records. *Journal of Physics A: Mathematical and Theoretical* 41, 22, 224015.

M. Castro-Neto, Y. S. Jeong, M. K. Jeong, and L. D. Han. 2009. Online-SVR for short-term traffic prediction under typical and atypical traffic conditions. *Expert Systems with Applications* 36, 3, 6164–6173.

P. S. Castro, D. Zhang, C. Chen, S. Li, and G. Pan. 2013. From taxi GPS traces to social and community dynamics: A survey. *ACM Computer Survey* 46, 2, Article 17, 34 pages.

I. Ceapa, C. Smith, and L. Capra. 2012. Avoiding the crowds: Understanding tube station congestion patterns from trip data. In *Proceedings of the 1st ACM SIGKDD International Workshop on Urban Computing*. ACM, 134–141.

V. Chandola, A. Banerjee, and V. Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys* 41, 3, 1–58.

S. Chawla, Y. Zheng, and J. Hu. 2012. Inferring the root cause in road traffic anomalies. In *Proceedings of the 2012 IEEE International Conference on Data Mining.* IEEE, 141–150.

X. Chen, Y. Zheng, Y. Chen, Q. Jin, W. Sun, E. Chang, and W. Y. Ma. 2014. Indoor air quality monitoring system for smart buildings. In *Proceedings of the 16th ACM International Conference on Ubiquitous Computing*. ACM.

Y. Chen, K. Jiang, Y. Zheng, C. Li, and N. Yu. 2009. Trajectory simplification method for location-based social networking services. In *Proceedings of the 1st ACM GIS Workshop on Location-based Social Networking Services*. ACM, 33–40.

Z. Chen, H. T. Shen, X. Zhou, Y. Zheng, and X. Xie. 2010. Searching trajectories by locations: An efficiency study. In *ACM SIGMOD International Conference on Management of Data.* ACM, 255–266.

J. Cranshaw, R. Schwartz, J. Hong, and N. Sadeh. 2012 The livehoods project: Utilizing social media to understand the dynamics of a city. *Association for the Advancement of Artificial Intelligence.*

J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. 2010. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international Conference on Ubiquitous Computing*. ACM, 119–128.

S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath. 2013. Real-time air quality monitoring through mobile sensing in metropolitan areas. In *Proceeding of the 2nd ACM SIGKDD International Workshop on Urban Computing*. ACM.

E. D'Hondt and S. Matthias. 2011. Participatory noise mapping. In *Proceedings of the 9th International Conference on Pervasive Computing.* Springer, 33–36.

D. Douglas and T. Peucker. 1973. Algorithms for the reduction of the number of points required to represent a line or its caricature. *Canadian Cartographer* 10, 2, 112–122.

I. Dusparic, C. Harris, A. Marinescu, V. Cahill, and S. Clarke. 2013. Multi-agent residential demand response based on load forecasting. In *Proceedings of the IEEE Conference on Technologies for Sustainability – Engineering and the Environment.*

E. Galvan-Lopez, A. Taylor, S. Clarke, and V. Cahill. 2014. Design of an automatic demand-side management system based on evolutionary algorithms. In *Proceedings of the the 29th Annual ACM Symposium on Applied Computing*, ACM, 24–28.

R. Gandia. 2012. City outlines travel diary plan to determine future transportation needs. *Calgary Sun*. Available at http://www.calgarysun.com/2012/01/11/city-outlines-travel-diary-plan-to-determine-future-transportation-needs.

Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani. 2010. An energy-efficient mobile recommender system. In *Proceedings of 16th SIGKDD Conference on Knowledge Discovery and Data Mining.* ACM, 899–908.

F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. 2007. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 330–339.

D. Goldberg, N. David, M. O. Brain, and T. Douglas. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35, 12, 61–70.

J. Goldman, K. Shilton, J. Burke, D. Estrin, M. Hansen, N. Ramanathan, S. Reddy, V. Samanta, M. Srivastava, and R. West. 2009. Participatory sensing: A citizen-powered approach to illuminating the patterns that shape our world. White paper.

M. C. González, C. A. Hidalgo, and A. L. Barabási. 2008. Understanding individual human mobility patterns. *Nature* 453, 7196, 779–782.

J. Gudmundsson and M. V. Kreveld. 2006. Computing longest duration flocks in trajectory data. In *Proceedings of the 14th International Conference on Advances in Geographical Information Systems*. ACM, 35–42.

J. Gudmundsson, M. V. Kreveld, and B. Speckmann. 2004. Efficient detection of motion patterns in spatio-temporal data sets. In *the Proceedings of the 12th International Conference on Advances in Geographical Information Systems*. ACM, 250–257.

A. Guehnemann, R. P. Schaefer, K. U. Thiessenhusen, and P. Wagner. 2004. *Monitoring Traffic and Emissions by Floating Car Data*. Institute of Transport Studies, Australia.

C. Faloutsos, T. G. Kolda, and J. Sun. 2007. Mining large time-evolving data using matrix and tensor tools. In *Proceedings of the ICML 2007*.

J. Froehlich, J. Neumann, and N. Oliver. 2009. Sensing and predicting the pulse of the city through shared bicycling. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. 1420–1426.

Y. Fu, H. Xiong, Y. Ge, Z. Yao, and Y. Zheng. 2014. Exploiting geographic dependencies for real estate appraisal: A mutual perspective of rand clustering. In *Proceedings of the 20th SIGKDD conference on Knowledge Discovery and Data Mining*. ACM.

M. Haklay and P. Weber. 2008. Openstreetmap: User-generated street maps. *Pervasive Computing* 7, 4, 12–18.

S. Hanson and P. Hanson. 1980. Gender and urban activity patterns in Uppsala, Sweden. *Geographical Review* 70, 3, 291–299.

C. Harris, R. Doolan, I. Dusparic, A. Marinescu, V. Cahill, and S. Clarke. 2014. A distributed agent based mechanism for shaping of aggregate demand. In *Proceedings of ENERGYCON 2014.*

J. C. Herrera, D. Work, X. Ban, R. Herring, Q. Jacobson, and A. Bayen. 2010. Evaluation of traffic data obtained via GPS-enabled mobile phones: The Mobile Century field experiment. *Transportation Research C* 18, 568–583.

R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen. 2010. Estimating arterial traffic conditions using sparse probe data. In *Proceedings of the 13th IEEE International Conference on Intelligent Transportation Systems.* IEEE, 923–929.

V. J. Hodge and J. Austin. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review* 22, 2, 85–126.

L. Hong, Y. Zheng, D. Yung, J. Shang, and L. Zou. 2014. Detecting black holes and volcanoes in a spatio-temporal graph. Submitted to ICDE 2015.

D. Hristova, G. Quattrone, A. Mashhadi, and L. Capra. 2013. The life of the party: Impact of social mapping in OpenStreetMap. In *Proceedings of the 7th AAAI Conference on Weblogs and Social Media*. AAAI Press.

C. C. Hung, C. W. Chang, and W. C. PENG. 2009. Mining trajectory profiles for discovering user communities. In *Proceedings of the 1st ACM SIGSPATIAL GIS Workshop on Location Based Social Networks*. ACM, 1–8.

T. Hunter, R. Herring, P. Abbeel, and A. Bayen. 2009. Path and travel time inference from GPS probe vehicle data. In *Proceedings of the International Workshop on Analyzing Networks and Learning with Graphs*.

H. Jeung, M. Yiu, X. Zhou, C. Jensen, and H. Shen. 2008a. Discovery of convoys in trajectory databases. *Proceedings of the VLDB Endowment* 1, 1, 1068–1080.

H. Jeung, H. Shen, and X. Zhou. 2008b. Convoy queries in spatio-temporal databases. In *Proceedings of the 24th International Conference on Data Engineering.* IEEE, 1457–1459.

S. Jiang, G. Fiore, Y. Yang, J. Ferreira, E. Frazzoli, and M. C. González. 2013. A review of urban computing for mobile phone traces: Current methods, challenges and opportunities. In *Proceedings of the 2nd SIGKDD Workshop on Urban Computing*.

S. Jiang, J. Ferreira, and M. C. Gonzalez. 2012. Discovering urban spatial-temporal structure from human activity patterns. In *Proceedings of the 1st ACM SIGKDD International Workshop on Urban Computing*. ACM, 95–102.

A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs. 2010. Urban cycles and mobility patterns: exploring and predicting trends in a bicycle-based public transport system. *IEEE Pervasive and Mobile Computing* 6, 455–466.

E. Kanoulas, Y. Du, T. Xia, and D. Zhang. 2006. Finding fastest paths on a road network with speed patterns. In *Proceedings of the 22nd International Conference on Data Engineering*.

D. Karamshuk, A. Noulas, S. Scellato, V. M. Nicosia, and Cecilia. 2013. Geo-spotting: Mining online location-based services for optimal retail store placement. In *Proceedings of the 19th ACM International Conference on Knowledge Discovery and Data Mining*. ACM, 793–801.

E. Keogh, J. Chu, S. D. Hart, and M. J. Pazzani. 2001. An on-line algorithm for segmenting time series. In *Proceedings of the International Conference on Data Mining.* IEEE, 289–296.

T. Kindberg, M. Chalmers, and E. Paulos. 2007. Gest editors' introduction: Urban computing. *Pervasive Computing* 6, 3, 18–20.

T. G. Kolda and B. W. Bader. 2009. Tensor decompositions and applications. *SIAM Review* 51, 3, 455–500.

V. Kostakos and E. O'Neill. 2008. Cityware: Urban computing to bridge online and real-world social networks. In *Handbook of Research on Urban Informatics*. Information Science Reference, Hershey, PA.

J. Krumm and E. Horvitz. 2006. Predestination: Inferring destinations from partial trajectories. In *Proceedings of the 8th International Conference on Ubiquitous Computing.* ACM, 243–260.

N. Lathia and L. Capra. 2011a. Mining mobility data to minimise travellers' spending on public transport. In *Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 1181–1189.

N. Lathia and L. Capra. 2011b. How smart is your smartcard? Measuring travel behaviours, perceptions, and incentives. In *Proceedings of the 13th ACM International Conference on Ubiquitous Computing*. ACM, 291–300.

S. A. Lathia and L. Capra. 2012. Measuring the impact of opening the London shared bicycle scheme to casual users. *Transportation Research Part C* 22, 88–102.

N. Lathia, J. Froehlich, and L. Capra. 2010. Mining public transport usage for personalised intelligent transport systems. In *Proceedings of the 10th IEEE International Conference on Data Mining*. IEEE, 887–892.

R. Lee and K. Sumiya. 2010. Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection. In *Proceedings of ACM SIGSPATIAL GIS Workshop on Location Based Social Networks*. ACM, 1–10.

D. Lee, H. Wang, R. Cheu, and S. Teo. 2004. Taxi dispatch system based on current demands and real-time traffic conditions. *Transportation Research Record: Journal of the Transportation Research Board*, 1882(-1):193–200.

Z. Li, B. Ding, J. Han, and R. Kays. 2010. Swarm: Mining relaxed temporal moving object clusters. *Proceedings of the VLDB Endowment* 3, 1–2, 723–734.

Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W. Ma. 2008. Mining user similarity based on location history. In *Proceedings of the 16th International Conference on Advances in Geographic Information System.* ACM.

W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xie. 2011. Discovering Spatio-Temporal Causal Interactions in Traffic Data Streams. In *Proceedings of 17th SIGKDD conference on Knowledge Discovery and Data Mining*. ACM Press:

B. Liu, Y. Fu, Z. Yao, and H. Xiong. 2013. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM.

Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. 2009. Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th ACM SIGSPATIAL Conference on Geographical Information Systems*. ACM, 352–361.

G. Lukasz and Ö. M. Tamer. 2010. *Data Stream Management*. Morgan and Claypool.

S. Ma, Y. Zheng, and O. Wolfson. 2013. T-Share: A large-scale dynamic taxi ridesharing service. In *Proceedings of IEEE International Conference on Data Engineering.* IEEE.

N. Maratnia and R. A. de By. 2004. Spatio-temporal compression techniques for moving point objects. In *Proceedings of the 9th International Conference on Extending Database Technology.* IEEE, 765–782.

A. Mashhadi, G. Quattrone, and L. Capra. 2013. Putting ubiquitous crowd-sourcing into context. In *Proceedings of the 16th ACM Conference on Computer Supported Cooperative Work and Social Computing*. ACM, 611–622.

A. Marinescu, I. Dusparic, C. Harris, S. Clarke, and V. Cahill. 2014. A dynamic forecasting method for small scale residential electrical demand. In *Proceedings of the International Joint Conference on Neural Networks.* IEEE.

D. Martinoc, S. M. Bertolottoa, F. Ferruccic, T. Kechadi, and P. Compieta. 2007. Exploratory spatio-temporal data mining and visualization. *Journal of Visual Languages and Computing* 18, 3, 255–279.

M. Momtazpour, P. Butler, M. S. Hossain, M. Bozchalui, N. Ramakrishnan, and R. Sharma. 2013. Coordinated Clustering Algorithms to Support Charging Infrastructure Design for Electric Vehicles. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*.

A. Nakamura and N. Abe. 1998. Collaborative filtering using weighted majority prediction algorithms. In *Proceedings of the 15th International Conference on Machine Learning*. ACM, 395–403.

R. Nair, E. Miller-Hooks, R. Hampshire, and A. Busic. 2013. Large-scale bicycle sharing systems: Analysis of V'Elib. *International Journal of Sustainable Transportation* 7, 1, 85–106.

M. Nicolas, M. Stevens, M. E. Niessen, and L. Steels. 2009. NoiseTube: Measuring and mapping noise pollution with mobile phones. In *Information Technologies in Environmental Engineering.* Springer, Berlin, 215–228.

K. Nigam and R. Ghani. 2000. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the 9th International Conference on Information and Knowledge Management*. ACM, 86–93.

S. J. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transaction on Knowledge Discovery and Data Engineering*, 22, 10, 1345–1359.

B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi. 2013. Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21th ACM SIGSPATIAL Conference on Advances in Geographical Information Systems*. ACM.

K. Panciera, R. Priedhorsky, T. Erickson, and L. Terveen. 2010. Lurking? Cyclopaths? A quantitative lifecycle analysis of user behavior in a Geowiki. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 1917–1926.

L. X. Pang, S. Chawla, W. Liu, and Y. Zheng. 2011. On mining anomalous patterns in road traffic streams. In *Proceedings of the 7th International Conference on Advanced Data Mining and Applications*. Springer, 237–251.

L. X. Pang, S. Chawla, W. Liu, and Y. Zheng. 2013. On Detection of Emerging Anomalous Traffic Patterns Using GPS Data. *Data and Knowledge Engineering (DKE)* 87, 357–373.

S. Phithakkitnukoon, M. Veloso, C. Bento, A. Biderman, and C. Ratti. 2010. Taxi-aware map: Identifying and predicting vacant taxis in the city. In *Proceedings of the 1st International Joint Conference on Ambient Intelligence.* 86.

D. Pfoser. 2008. Floating car data. In *Encyclopedia of GIS*. Springer.

D. Pfoser, S. Brakatsoulas, P. Brosch, M. Umlauft, N. Tryfona, and G. Tsironis. 2008. Dynamic travel time provision for road networks. In *Proceedings of the 16th International Conference on Advances in Geographic Information Systems*. ACM.

R. Priedhorsky, M. Masli, and L. Terveen. 2010. Eliciting and focusing geographic volunteer work. In *Proceedings of the 13th ACM Conference on Computer Supported Cooperative Work and Social Computing.* ACM, 61–70.

R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu. 2013. Ear-phone: A context-aware noise mapping using smart phones. In eprint arXiv:1310.4270.

R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu. 2010. Ear-phone: An end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 105–116.

C. Ratti, S. Sobolevsky, F. Calabrese, C. Andris, J. Reades, M. Martino, R. Claxton, and S. H. Strogatz. 2010. Redrawing the map of Great Britain from a network of humani. *PLoS ONE* 5, 12.

S. Rinzivillo, S. Mainardi, F. Pezzoni, M. Coscia, D. Pedreschi, and F. Giannotti. 2012. Discovering the geographical borders of human mobility. *Künstl Intell* 26, 253–260.

R. Rro. 1997. PARAFAC. Tutorial and applications. In *Chemometrics and Intelligent Laboratory Systems* 38, 2, 149–171.

J. Shang, Y. Zheng, W. Tong, and E. Chang. 2014. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proceedings of 20th SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.

S. Shaheen, S. Guzman, and H. Zhang. 2010. Bikesharing in Europe, the Americas, and Asia: Past, present, and future. In: *2010 Transportation Research Board Annual Meeting*. Washington, DC, USA.

C. Sheng, Y. Zheng, W. Hsu, M. L. Lee, and X. Xie. 2010. Answering top-k Similar Region Queries. In *Proceedings of the Database Systems for Advanced Applications*. Springer, 186–201.

S. Silvia, B. Ostermaier, and A. Vitaletti. 2008. First experiences using wireless sensor networks for noise pollution monitoring. In *Proceedings of the Workshop on Real-World Wireless Sensor Networks*. ACM, 61–65.

R. Song, W. Sun, B. Zheng, Y. Zheng, C. Tu, and S. Li. 2014. PRESS: A novel framework of trajectory compression in road networks. In *Proceedings of 40th International Conference on Very Large Data Bases*.

X. Song, Q. Zhang, Y. Sekimoto, Horanont, T. S. Ueyama, and R. Shibasaki. 2013. Modeling and probabilistic reasoning of population evacuation during large-scale disaster. In *Proceedings of the 19th SIGKDD Conference on Knowledge Discovery and Data Mining.* ACM, 1231–1239.

Y. Sun, R. Zhang, and Y. Zheng. 2014. Spatial aggregate queries in location-based social networks. Submitted to SIGMOD.

L. A. Tang, Y. Zheng, X. Xie, J. Yuan, X. Yu, and J. Han. 2011. Retrieving k-nearest neighboring trajectories by a set of point locations. In *Proceedings of the 12th Symposium on Spatial and Temporal Databases*. Volume 6849, Springer, 223–241.

L. A. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, W.-C. Peng, T. L. Porta, and L. Kaplan. 2013. A framework of traveling companion discovery on trajectory data streams. *ACM Transaction on Intelligent Systems and Technology* 2013.

L. A. Tang, Y. Zheng, J. Yuan, J. Han, A. Leung, C. C. Hung, and W.C. Peng. 2012. Discovery of traveling companions from streaming trajectories. In *Proceedings of the 28th IEEE International Conference on Data Engineering*. IEEE, 186–197.

Y. Theodoridis, M. Vazirgiannis, and T. K. Sellis. 1996. Spatio-temporal indexing for large multimedia applications. In *Proceedings of the 3rd International Conference on Multimedia Computing and Systems*. IEEE, 441–448.

A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. 2009. VTrack: Accurate, energy-aware road traffic delay estimation using mobile phones. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems.*

J. L. Toole, M. Ulm, M. C. González, and D. Bauer. 2012. Inferring land use from mobile phone activity. In *Proceedings of the ACM SIGKDD International Workshop on Urban Computing.* ACM, 1–8.

J. Tulusan, T. Staake, and E. Fleisch. Providing eco-driving feedback to corporate car drivers: What impact does a smartphone application have on their fuel efficiency? In *Proceedings of the 14th ACM Conference on Ubiquitous Computing.* ACM, 212–215.

S. Wakamiya, R. Lee, and K. Sumiya. 2012. Crowd-sourced urban life monitoring: Urban area characterization based crowd behavioral patterns from Twitter. In *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*. Article No. 26.

M. Wand and M. Jones. 1995. *Kernel Smoothing*, volume 60. Chapman and Hall/CRC.

L. Wang, Y. Zheng, X. Xie, and W. Y. Ma. 2008. A flexible spatio-temporal indexing scheme for large-scale GPS track retrieval. In *Proceedings of the 9th International Conference on Mobile Data Management.* IEEE, 1–8.

L. Wang, Y. Zheng, and Y. Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM.

K. Watkins, B. Ferris, A. Borning, S. Rutherford, and D. Layton. 2011. Where is my bus? Impact of mobile real-time information on the perceived and actual wait time of transit riders. *Transportation Research Part A* 45, 839–848.

L. Y. Wei, Y. Zheng, and W. C. Peng. 2012. Constructing popular routes from uncertain trajectories. In *Proceedings of the 18th SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 195–203.

X. Xiao, Y. Zheng, Q. Luo, and X. Xie. 2010. Finding similar users using category-based location history. In *Proceedings of the 18th ACM SIGSPATIAL Conference on Advances in Geographical Information Systems*. ACM, 442–445.

X. Xiao, Y. Zheng, Q. Luo, and X. Xie. 2012. Inferring social ties between users with human location history. *Journal of Ambient Intelligence and Humanized Computing*. December 5, 1, 3–19.

X. Xu, J. Han, and W. Lu. 1999. RT-tree: An improved R-tree index structure for spatio-temporal databases. In *Proceedings of the 4th International Symposium on Spatial Data Handling.* 1040–1049.

A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang, and Z. Xu. 2013. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction. In *Proceedings of the 29th IEEE International Conference on Data Engineering*. IEEE, 254–265.

K. Yamamoto, K. Uesugi, and T. Watanabe. 2010. Adaptive routing of cruising taxis by mutual exchange of pathways. *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5178. Springer, 559–566.

M. Ye, Y. Yin, W. Q. Lee, and D. L. Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang. 2011. Geographical topic discovery and comparison. In *Proceedings of the 20th International Conference on World Wide Web*. ACM, 247–256.

H. Yoon, Y. Zheng, X. Xie, and W. Woo. 2010. Smart itinerary recommendation based on user-generated GPS trajectories. In *Proceedings of the 7th Ubiquitous Intelligence and Computing*. 6406. Springer, 19–34.

H. Yoon, Y. Zheng, X. Xie, and W. Woo. 2011. Social itinerary recommendation from user-generated digital trails. *Journal on Personal and Ubiquitous Computing* 16, 5, 469–484.

N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie. 2014. T-Finder: A recommender system for finding passengers and vacant taxis. *IEEE Transactions on Knowledge and Data Engineering*.

J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun. 2011a. Where to find my next passenger? In *Proceedings of 13th ACM International Conference on Ubiquitous Computing*. ACM, 109–118.

J. Yuan, Y. Zheng, and X. Xie. 2012a. Discovering regions of different functions in a city using human mobility and POIs. In *Proceedings of 18th SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 186–194.

N. J. Yuan, Y. Zheng, and X. Xie. 2012b. Segmentation of urban areas using road networks. MSR-TR-2012-65.

J. Yuan, Y. Zheng, X. Xie, and G. Sun. 2011b. Driving with knowledge from the physical world. In *Proceedings of 17th SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 316–324.

N. J. Yuan, F. Zhang, D. Lian, K. Zheng, S. Yu, and X. Xie. 2013a. We know how you live: Exploring the spectrum of urban lifestyles. In *Proceedings of the ACM Conference on Online Social Networks.* ACM, 3–14.

J. Yuan, Y. Zheng, X. Xie, and G. Sun. 2013b. T-Drive: Enhancing driving directions with taxi drivers' intelligence. *Transactions on Knowledge and Data Engineering* 25, 1, 220–232.

J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. 2010a. T-Drive: Driving directions based on taxi trajectories. In *Proceedings of ACM SIGSPATIAL Conference on Advances in Geographical Information Systems*. ACM, 99–108.

J. Yuan, Y. Zheng, C. Zhang, X. Xie, and G. Sun. 2010b. An interactive-voting based map matching algorithm. In *Proceedings of the International Conference on Mobile Data Management*. IEEE Press, 43–52.

F. Zhang, D. Wilkie, Y. Zheng, and X. Xie. 2013. Sensing the pulse of urban refueling behavior. In *Proceedings of the 15th International Conference on Ubiquitous Computing.* ACM, 13–22.

Y. Zheng. 2012a. Tutorial on location-based social nNetworks. In *Proceedings of International conference on World Wide Web*.

Y. Zheng. 2011a. Location-based social networks: Users. In *Computing with Spatial Trajectories,* Y. Zheng and X. Zhou, Eds. Springer, 243–276.

W. C. Zheng, B. Cao, Y. Zheng, X. Xie, and Q. Yang. 2010a. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 236–241

Y. Zheng, X. Chen, Q. Jin, Y. Chen, X. Qu, X. Liu, E. Chang, W.-Y. Ma, Y. Rui, and W. Sun. 2014a. *A Cloud-Based Knowledge Discovery System for Monitoring Fine-Grained Air Quality.* MSR-TR-2014-40.

Y. Zheng, Y. Chen, Q. Li, X. Xie, and W. Y. Ma. 2010b. Understanding transportation modes based on GPS data for web applications. *ACM Transactions on the Web* 4, 1, 1–36.

Y. Zheng, Y. Chen, X. Xie, and W. Y. Ma. 2009a. GeoLife2.0: A location-based social networking service. In *Proceedings of International Conference on Mobile Data Management 2009.* IEEE, 357–358.

Y. Zheng, X. Feng, X. Xie, S. Peng, and J. Fu. 2010c. Detecting nearly duplicated records in location datasets. In *Proceedings of 18th ACM SIGSPATIAL Conference on Advances in Geographical Information Systems*. ACM, 137–143.

Y. Zheng and J. Hong. 2012b. *Proceedings of the 4th International Workshop on Location-Based Social Networks*. In conjunction with UbiComp 2012.

Y. Zheng, S. E. Koonin, and O. Wolfson. 2013a. In *Proceedings of the 2nd International Workshop on Urban Computing*. ACM.

Y. Zheng, Q. Li, Y. Chen, X. Xie, and W. Y. Ma. 2008a. Understand mobility based GPS data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*. ACM, 312–321.

Y. Zheng, L. Liu, L. Wang, and X. Xie. 2008b. Learning transportation mode from raw GPS data for geographic applications on the Web. In *Proceedings of the 11th International Conference on World Wide Web*. ACM, 247–256.

Y. Zheng, F. Liu, and H. P. Hsieh. 2013b. U-Air: When urban air quality inference meets big data. In *Proceedings of 19th SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 1436–1444.

Y. Zheng, T. Liu, Y. Wang, Y. Zhu, and E. Chang. 2014b. Diagnosing New York City's noises with ubiquitous data. In *Proceedings of the 16th International Conference on Ubiquitous Computing*. ACM.

Y. Zheng, Y. Liu, J. Yuan, and X. Xie. 2011b. Urban computing with taxicabs. In *Proceedings of the 13th International Conference on Ubiquitous Computing*. ACM, 89–98.

Y. Zheng and M. F. Mokbel. 2011c. *Proceedings of the 3rd International Workshop on Location-Based Social Networks*. In conjunction with ACM SIGSPATIAL GIS 2011.

Y. Zheng, L. Wang, R. Zhang, X. Xie, and W. Y. Ma. 2008c. GeoLife: Managing and understanding your past life over maps. In *Proceedings of the 9th International Conference on Mobile Data Management*. IEEE, 211–212.

Y. Zheng and O. Wolfson. 2012c. *Proceedings of the 1st International Workshop on Urban Computing*. In conjunction with KDD 2012.

Y. Zheng and X. Xie. 2010d. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin* 33, 2, 32–40.

Y. Zheng, X. Xie, and W. Y. Ma. 2008d. Search your life over maps. In *Proceedings of the International Workshop on Mobile Information Retrieval*. 24–27.

Y. Zheng and X. Xie. 2009b. Learning location correlation from GPS trajectories. In *Proceedings of the International Conference on Mobile Data Management.* IEEE, 27–32.

Y. Zheng and X. Xie. 2011d. Learning travel recommendations from user-generated GPS traces. *ACM Transactions on Intelligent Systems and Technology* 2, 1, 2–19.

Y. Zheng and X. Xie. 2011e. Location-based social networks: Locations. In *Computing with Spatial Trajectories,* Y. Zheng and X. Zhou, Eds. Springer, 277–308.

Y. Zheng, L. Zhang, X. Xie, and W. Y. Ma. 2009c. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the 18th International Conference on World Wide Web*. ACM, 791–800.

Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Y. Ma. 2010e. Recommending friends and locations based on individual location history. In *ACM Transactions on the Web* 5, 1, Article 5.

W. C. Zheng, Y. Zheng, X. Xie, and Q. Yang. 2010f. Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, 1029–1038.

W. C. Zheng, Y. Zheng, X. Xie, and Q. Yang. 2012d. Towards mobile intelligence: Learning from GPS history data for collaborative recommendation. *Artificial Intelligence Journal* 184–185, 17–37.

K. Zheng, Y. Zheng, N. J. Yuan, S. Shang, and X. Zhou. 2014. Online Discovery of Gathering Patterns over Trajectories. *IEEE Transactions on Knowledge Discovery and Engineering*.

K. Zheng, Y. Zheng, N. J. Yuan, and S. Shang. 2013b. On discovery of gathering patterns from trajectories. In *Proceedings of the 29th IEEE International Conference on Data Engineering*. IEEE, 242–253.

Y. Zheng and X. Zhou. 2011f. *Computing with Spatial Trajectories*. Springer.

Xiaojin. Zhu. 2008. *Semi-Supervised Learning Literature Survey.* Computer Sciences, University of Wisconsin-Madison (2008).

J. Zimmerman, A. Tomasic, C. Garrod, D. Yoo, C. Hiruncharoenvate, R. Aziz, N. R. Thiruvengadam, Y. Huang, and A. Steinfeld. 2011. Field trial of Tiramisu: Crowd-sourcing bus arrival times to spur co-design. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*. ACM, 1677–1686.

Received January 2014; revised April 2014; accepted March 2014