Chapter 43
AI and Deep Learning for Urban Computing

Senzhang Wang and Jiannong Cao

Abstract In the big data era, with the large volume of available data collected by various sensors deployed in urban areas and the recent advances in AI techniques, urban computing has become increasingly important to facilitate the improvement of people's lives, city operation systems, and the environment. In this chapter, we introduce the challenges, methodologies, and applications of AI techniques for urban computing. We first introduce the background, followed by listing key challenges from the perspective of computer science when AI techniques are applied. Then we briefly introduce the AI techniques that are widely used in urban computing, including supervised learning, semi-supervised learning, unsupervised learning, matrix factorization, graphic models, deep learning, and reinforcement learning. With the recent advances of deep-learning techniques, models such as CNN and RNN have shown significant performance gains in many applications. Thus, we briefly introduce the deep-learning models that are widely used in various urban-computing tasks. Finally, we discuss the applications of urban computing including urban planning, urban transportation, location-based social networks (LBSNs), urban safety and security, and urban-environment monitoring. For each application, we summarize major research challenges and review previous work that uses AI techniques to address them.

S. Wang (✉)
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China
e-mail: szwang@nuaa.edu.cn

J. Cao
Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
e-mail: csjcao@polyu.edu.hk

W. Shi et al. (eds.), *Urban Informatics*, The Urban Book Series,
https://doi.org/10.1007/978-981-15-8983-6_43

43.1 Background

In the big data era, sensing technologies (e.g., GPS and environment sensors) and large-scale computing infrastructures (e.g., distributed storage and computing) have produced and stored a variety of big data generated in urban space in real time, such as human-mobility data, air-quality data, transportation data, urban noise data, and urban crime data. Generally, big data can be defined as a field that studies the methodologies of effectively and efficiently storing, processing, extracting information from, discovering valuable knowledge from, and visualizing the datasets that are too large in data volume or too complex in data formats to be handled by traditional data storage, processing, and analytic paradigms. Usually, big data can be characterized by five Versus: volume, variety, velocity, veracity, and value (Ishwarappa and Anuradha 2015). The first primary characteristic of big data is its sheer volume. Variety means that the data can be unstructured, and the data types are much richer, including images, texts, videos, graphs, etc. As the data are usually generated in real-time and new data keep on coming, the characteristic of velocity requires that the new streaming data can be processed in near real time. Veracity refers to the trustworthiness of the data. Big data usually also mean big noise, such as in social-media data. The value hidden in the data can be low and may require carefully designed machine-learning or data-mining methods to discover useful knowledge from the massive data.

Mining knowledge hidden in the big data generated in urban areas is critically important to facilitate many real applications for smart cities, including relieving traffic congestion, urban crime prediction, real-time air pollution monitoring, urban planning, etc. To this aim, artificial intelligence (AI) techniques are urgently needed for knowledge discovery from the large-volume, noisy, heterogeneous, and ever-growing urban data (Zheng et al. 2014a, b). Recently, AI techniques driven by big data, such as the popular deep-learning models, have been widely used to solve diverse urban-computing tasks and have achieved success (Wang et al. 2019, 2020). For example, urban-traffic prediction and navigation driven by AI have been widely explored and applied in many applications such as the Gaode map for navigating and the City Brain system developed by Alibaba (Zhang et al. 2019a, b). As an interdisciplinary research field, knowledge discovery from urban big data is an indispensable part of urban computing, and AI techniques play a critically important role in mining correlations and patterns and predicting trends from the data.

Figure 43.1 shows a general framework to illustrate how AI techniques, especially machine learning, are used for various applications in urban computing. As shown in Fig. 43.1, there are three phases in general. The first phase is data acquisition. Diverse types of data generated from various sensors deployed in different locations in a city are collected, including GPS position data, air-quality data, weather data, data on social relations, points of interest (POIs), transportation networks, and social events. The collected raw data usually need to be preprocessed for further analysis.

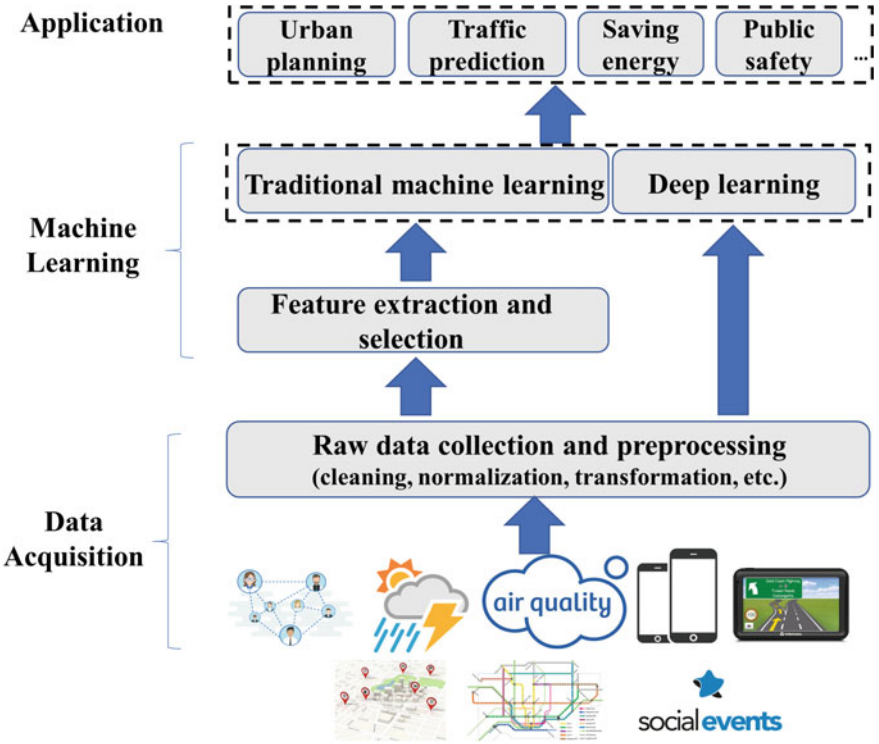


Fig. 43.1 Framework of applying AI techniques for urban computing

The data preprocessing operations include data cleaning, normalization, transformation, and instance selection. Next, the machine-learning phase performs pattern learning or knowledge discovery from the data. For traditional machine-learning methods, features need to be first extracted and selected from the data manually through feature engineering. In machine learning, features refer to a set of measurable properties or characteristics of the objects under study. They are used as the input of the machine-learning algorithms to be mapped to the output. Discriminating features can be extracted and selected from the raw data based on domain knowledge, and then fed into a machine-learning model such as the SVM classifier or logistic regression for training. Note that for the deep-learning models that are extremely popular nowadays, they do not need handcrafted features. Deep-learning models can automatically learn features from the raw data and integrate the feature learning and model learning in an end-to-end way, which is a significant advantage. The third phase is using the trained machine-learning models to support various urban-computing applications, such as urban planning, traffic prediction, public safety, and energy saving. The results of machine-learning models can provide us with knowledge, predictions, and guidance to help us make decisions on how to build a smarter city.

In the remainder of the chapter, we first present the challenges of using AI techniques for analyzing and discovering knowledge from urban data. Then, we introduce both traditional AI models and recent deep-learning models that are widely used in various tasks of urban computing. Next, we classify urban computing into several application categories and review-related work, respectively.

43.2 Challenges

Compared with other types of data, there are some unique challenges for conducting machine learning using the big data generated from various urban sensors.

Data acquisition: Usually, a large number of sensors should be deployed in different locations of a city for data collection. However, there are several reasons why the sensors cannot be massively deployed all around the city. First, some sensors are expensive, such as cameras and sensors in air-quality monitoring stations. Second, due to the energy consumption constraint, the number of sensors is usually limited. Sometimes it is difficult to select suitable locations to deploy sensors for data acquisition. It is also nontrivial to estimate the data at a location where there are no sensor readings, based on the observed sensor data from other locations.

Large volume and streaming data: The volume of the data generated from an urban area is usually very large considering the large number of sensors deployed in a city; and the data volume grows quickly, considering that the sensors generate data continuously in real time. Traditional machine-learning or data-mining techniques usually need a large number of labeled training samples and thus are time consuming. Many urban-computing tasks need real-time data analysis, such as traffic prediction and air-quality monitoring. Therefore, it is challenging for existing AI techniques to process this large volume of data continuously and almost instantly.

Heterogeneous data: Solving a specific task in urban computing usually involves multiple datasets rather than only one dataset. For example, city-wide air-pollution prediction involves the simultaneous study of multiple types of data, including traffic flow, weather, and land uses. Different datasets usually present diverse data formats or types. Traditional data-mining and machine-learning techniques are usually designed to handle one type of data, such as image, text, and graphics. How to fuse the heterogeneous data with different formats and structures involved in one learning task to serve the urban-computing application of interest is difficult, and also a hot research topic currently.

Complex dependencies among the data: Different types of urban data can be highly correlated, such as traffic data, air-quality data, and weather data. Traffic congestion is usually highly correlated with POI distribution, time of day, and social events. It is difficult for traditional statistics-based methods to capture the correlations and dependencies among the data without the help of domain expertise. Mining the dependencies among the data may be especially important to help improve various urban-computing applications such as urban planning, policy making, and intelligent transportation systems.

Noisy and incomplete data: Most data in urban computing are generated by urban sensors which are deployed in an open environment (e.g., the air-quality sensors deployed on the field). The sensors may fail to work normally and produce wrong or noisy data from time to time. In addition, some sensors are expensive, and only a limited number of sensors are deployed due to this cost limitation. For example, the road cameras for traffic monitoring are usually only installed in some intersections of a road network due to the high cost. Performing a task such as city-wide air-quality and traffic monitoring with such noisy and incomplete data is challenging.

Distributed data storage and processing: As the urban sensors are deployed at different locations, and the data volume increases rapidly, a distributed data-storage and processing infrastructure is usually required for more efficient computation of various machine-learning and data-mining algorithms. Considering the heterogeneity of the urban data, the complex dependencies among the data, and the nonuniform distributions of the data sensors, it is very challenging to design such a distributed data-storage and processing infrastructure.

Data privacy: Urban data are mostly collected from users. For example, users' mobility data can be collected from users' smartphones, and the urban-traffic data can be collected from the GPS module installed in private vehicles. How to protect the data privacy of the users and at the same time use the data to facilitate various applications such as navigation and travel route recommendation is a non-trivial problem. There needs a tradeoff between data privacy and data utility (see Chap. 32).

To address the above-mentioned challenges, various AI techniques are being explored in different application scenarios of urban computing, such as supervised learning, semi-supervised learning, unsupervised learning, matrix factorization, graphic models, deep learning, and reinforcement learning. Next, we briefly introduce the concept and preliminary knowledge of the methods and then discuss how these models can be used in different tasks of urban computing in detail.

43.3 Traditional AI Techniques

43.3.1 Supervised Learning

Supervised learning, such as classification and regression, is a type of machine learning that learns a function mapping the input features to an output label or variable, based on a set of training input–output pairs (Caruana and Niculescu Mizil 2006). Note that in supervised learning, a training dataset that contains both the input data and the corresponding output labels or variables is needed, and the goal is to learn a mapping function from the training dataset.

Supervised learning is widely used in many urban-computing tasks when a large number of labeled training data samples are available, such as traffic prediction

(Castro-Neto et al. 2009), region classification (Toole et al. 2012), and POI recommendation (Daniel and Sebastian 2000). For example, Toole et al. (2012) studied the problem of inferring the types of urban land-use from users' mobile-phone activity data. A supervised classification algorithm was used to identify four types of land uses with similar zoned uses and mobile-phone activity patterns. The training data of the algorithm contained three weeks of call records for about 600,000 users in the Boston region. Castro-Neto et al. (2009) proposed a supervised regression algorithm called online support vector machine to predict short-term freeway traffic flow under both typical and atypical conditions.

43.3.2 Unsupervised Learning

Significantly different from supervised learning, unsupervised learning does not need any labeled data for training. Unsupervised learning aims to capture the underlying structures, patterns, or distributions from the input data without the guidance of output labels or variables. Unsupervised learning can be generally grouped into clustering and association. Clustering is the task of grouping a set of objects so that objects in the same group are more similar to each other than to those in other groups. Each object group is called a cluster. Association-rule learning is a rule-based machine-learning method for discovering interesting relations between variables or patterns in large databases. Association-rule learning algorithms intend to identify such strong rules or patterns in the given dataset using measures of interestingness.

In many real application scenarios, there are no labeled data at all. In such a case, unsupervised learning techniques can be used for mining knowledge from the massive data. For example, mining patterns from the trajectories of moving objects is an important research topic in spatial–temporal data mining (Giannotti et al. 2007). There are no labeled training data for discovering new patterns in trajectories, and thus the unsupervised pattern-mining methods are applied. Another example is city-boundary detection driven by big data. This task aims to discover the real borders of a city according to the interactions between people, using GPS tracks or phone-call records, and there are no ground-truth labels for the boundary of a city. To solve this problem, Rinzivillo et al. (2012) proposed to first build a location network based on human interaction and then partition the network using an unsupervised community-detection method. The boundaries of regions can be thus characterized by the discovered location clusters, with denser interaction between locations in the cluster.

43.3.3 Semi-supervised Learning

Semi-supervised learning falls between unsupervised learning, which does not have labeled training data at all, and supervised learning which has complete labeled

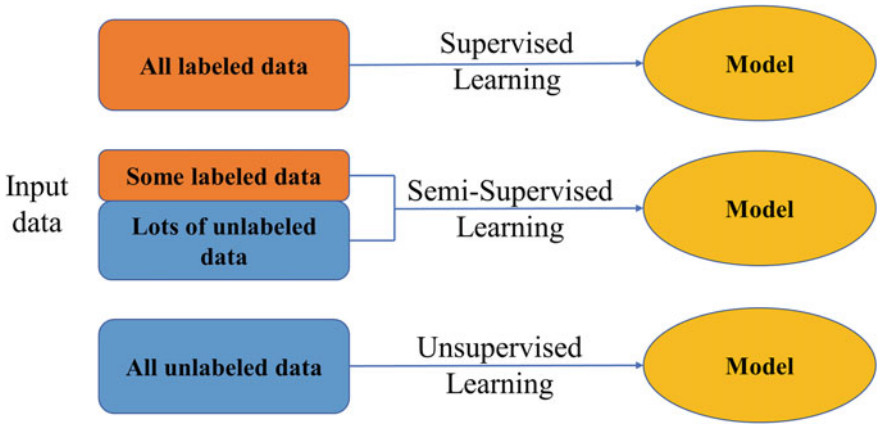


Fig. 43.2 Three types of machine-learning methods

training data. Semi-supervised learning makes use of both a small amount of available labeled data and a large amount of unlabeled data for training (Zhu 2005). As it is usually expensive and time-consuming to label a large number of training data for supervised learning, semi-supervised learning is widely used based on the observation that unlabeled data, when used in conjunction with a small amount of labeled data, can achieve considerable performance improvement over unsupervised learning. Semi-supervised learning also has broad applications in urban computing. For example, Zheng et al. (2013) proposed a semi-supervised learning approach based on a co-training framework to predict the air quality of a location where there is no air-quality monitoring station already. The used co-training framework consisted of two separated classifiers, with one using spatially related features and the other using temporally related features. Figure 43.2 compares three types of machine-learning methods.

43.3.4 Matrix Factorization

Matrix factorization, which is also called matrix decomposition, decomposes a matrix into a product of two or three smaller matrices. It is an approach that can simplify some complex matrix operations, since these can be performed on the decomposed smaller matrices rather than on the original large matrix (Daniel and Sebastian 2000). Popular matrix factorization methods include LU decomposition, QR decomposition, Jordan decomposition, and SVD. From an application point of view, matrix factorization can be used to discover the latent features underlying the interactions between two types of entities, such as users and items in recommendation systems. For example, SVD is widely used in collaborative filtering (Zhou et al. 2015), which factorizes the product-rating matrix A into the product of three smaller matrices, the left singular

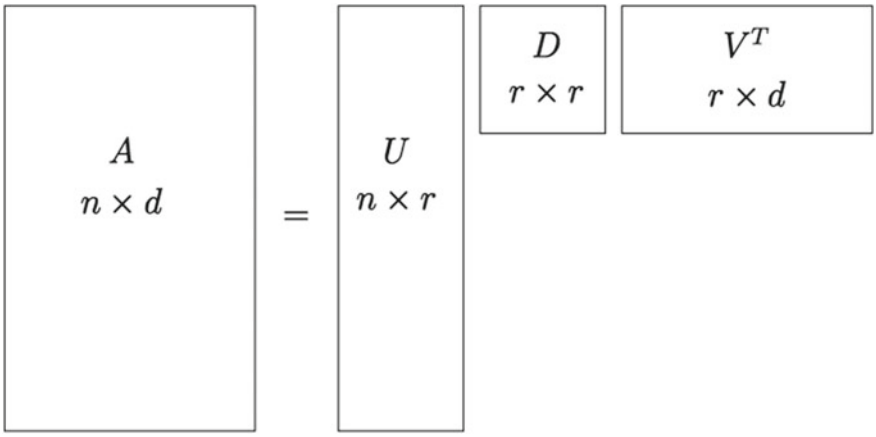


Fig. 43.3 Illustration of SVD

vectors U, the singular values D, and the right singular vectors V^T as shown in Fig. 43.3. Matrix factorization has very broad applications in machine learning, such as image processing, data compression, spectral clustering, recommendation, and matrix completion. For example, when the original matrix A is incomplete, with many unknown entry values, we can approximate it with three factorized low-rank matrices and estimate the missing entries in A to complete it.

Matrix factorization is widely used in many estimation or inference-related urban-computing tasks such as location recommendation, urban noise estimation, and urban-traffic estimation. For example, Zheng et al. (2010) proposed to collaboratively recommend location and activity to users through factorizing the location-activity matrix constructed from users' GPS historical trajectory data. Zheng et al. (2014a, b) integrated tensor composition and matrix composition to infer the fine-grained noise distribution at different times of day for each region of NYC. The noise distribution of NYC was modeled with a three-dimension tensor, whose three dimensions are regions, noise categories, and time slots. Supplementing the missing entries of the noise distribution tensor using the proposed tensor-matrix co-factorization approach, the noise distribution throughout the entire NYC can be inferred. Wang et al. (2019, 2020) proposed a locally balanced inductive matrix factorization model to infer the bike usage of a city at different hours of the day for dockless bike-sharing systems. The bike usage demand was modeled as a matrix whose two dimensions are region ID and time slot, and the entries are the needed number of bikes. The unknown entries of the bike-demand matrix are inferred through a proposed inductive matrix factorization method.

43.3.5 Graphical Model

A graphical model uses a graph to express the conditional dependency relationships among different random variables and is also called the probabilistic graphical model (PGM; Koller and Friedman 2009). It is widely used in probability theory, Bayesian statistics, and machine learning. Generally, graphical models use a graph-based representation to encode the variable distributions over a multi-dimensional space, which provides a general framework for modeling large collections of random variables with complex interactions. There are two types of commonly used graphical representations of variable distributions: Bayesian networks and Markov random fields. Figure 43.4 shows an example of a simple graphical model. Each node in the graph denotes a variable, and each arrow indicates a dependency relationship between two variables. In this example, D depends on A, B, and C; and C depends on B and D; whereas A and B are independent to each other.

In many urban-computing tasks, the data can be heterogeneous and collected from different sources, and the interactions and correlations among the data are usually complex. Graphical models can be used to model the dependencies among the data and make accurate estimates or inference. For example, in urban-traffic estimation and prediction, the traffic conditions of a road segment can be affected by both the neighboring road segments and the external factors such as weather, holidays, and rush hours. Wang et al. (2016a, b) proposed to use a coupled hidden Markov model for road-network-level traffic-congestion estimation. In this model, the traffic condition of a road segment at time t depends on its previous traffic condition at $t - 1$ and the traffic conditions of its neighboring road segments at $t - 1$. To model the complex dependencies among them, a graphical model that uses multiple coupled Markov chains was proposed. Shang et al. (2014) studied the problem of instantly inferring the gas consumption and pollution emission of the vehicles traveling on a road network of a city, based on the GPS trajectory data collected from a sample

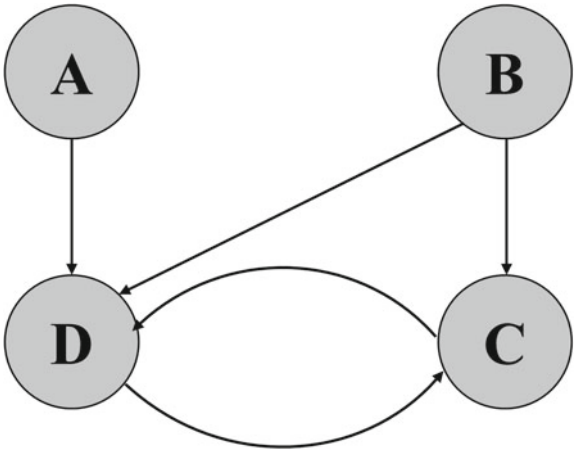


Fig. 43.4 A toy example of a graphical model

of vehicles. To address this task, they proposed an unsupervised dynamic Bayesian network model called the traffic volume inference model (TVI) to infer the number of vehicles passing each road segment per minute. TVI can model the effect of multiple external and internal factors on the traffic volume, including the travel speed, weather conditions, and the geographic features of a road.

43.4 Deep Learning

Deep learning is a type of machine-learning method whose structure, called an artificial neural network (ANN), is inspired by the structure and function of the human brain. The initial form of an artificial neural network is the perceptron, which was proposed in the 1950s (Rosenblatt 1957). Although ANNs have been proposed and studied for many years, early ANN models were not that successful compared with other machine-learning models, such as the Bayesian model and SVM, due to their shallow structures with only two or three layers of neurons. In recent years, ANN models with much deeper model structures containing tens of or even hundreds of neural layers are gaining popularity due to their supremacy in terms of prediction accuracy when trained with huge amounts of data (LeCun et al. 2015). Figure 43.5 shows the performance curves of deep-learning methods and most other traditional machine-learning methods with increasing amounts of training data. One can see that the learning performance of traditional methods first increases with an increase in the data amount and then reaches a performance bottleneck. More data will not lead to better performance due to the limited learning ability of traditional methods. For deep learning; however, the performance keeps on increasing with more and more training data, which is mainly due to its deep structure and powerful hierarchical feature-learning ability.

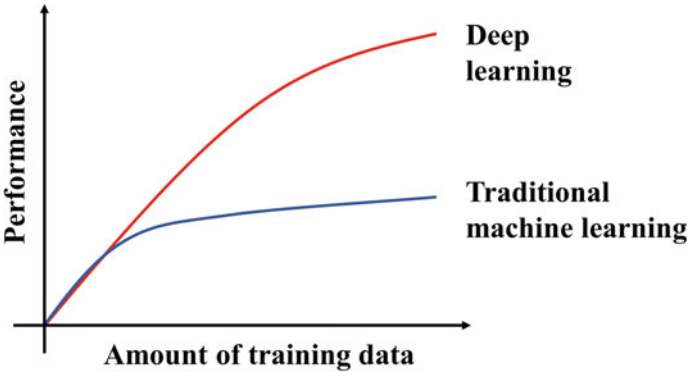


Fig. 43.5 Performance curves of deep learning and traditional machine learning with increasing amounts of training data

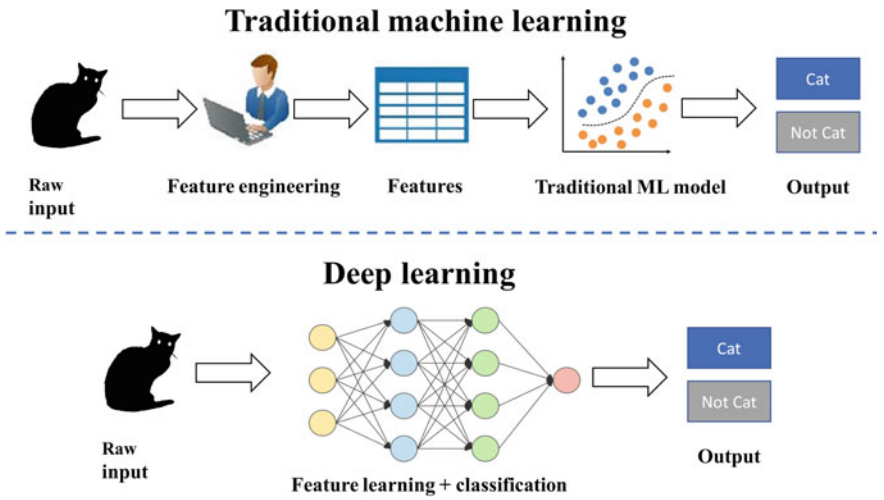


Fig. 43.6 Traditional machine learning *vs* deep learning

Besides the powerful learning ability from big data, another significant difference and advantage of deep learning compared with traditional machine learning is that deep learning does not need handcrafted features and can learn features from the input raw data automatically. Figure 43.6 shows a pipeline comparison between traditional machine learning and deep learning. We can see that for traditional machine-learning models, given the raw input data, feature engineering is first conducted to manually extract the features, and then, the features are input into the machine-learning model for classification. For deep-learning models, feature engineering is not needed any more. Feature learning and model learning are performed in an end-to-end learning way for deep-learning models.

Deep-learning architectures such as deep neural networks (DNN), deep belief networks (DBN), recurrent neural networks (RNN), and convolutional neural networks (CNN) have been widely applied in the fields of computer vision, speech recognition, natural-language processing, audio recognition, social-network analysis, machine translation, bioinformatics, medical-image analysis, and urban computing, where they have produced results comparable to and in some cases superior to humans. Next, we will briefly introduce some deep-learning models that are widely used in the tasks of urban computing.

43.4.1 Restricted Boltzmann Machines (RBM)

A restricted Boltzmann machine is a two-layer stochastic neural network (LeCun et al. 2015), which is broadly used for dimensionality reduction, classification, feature learning, and collaborative filtering. As shown in Fig. 43.7, RBM generally contains

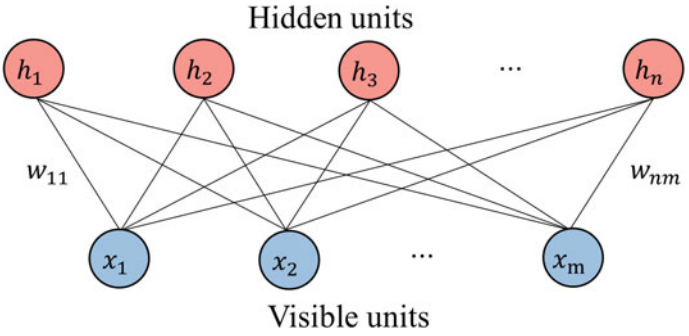


Fig. 43.7 Structure of RBM

two layers. The first layer of RBM is called the visible layer with the neuron nodes $\{x_1, x_2, \ldots, x_m\}$, and the second layer is the hidden layer with the neuron nodes $\{h_1, h_2, \ldots, h_n\}$. The structure of RBM can be considered as a fully connected bipartite undirected graph. All nodes in RBM are connected to each other across layers by undirected weight edges $\{w_{11}, w_{22}, \ldots, w_{nm}\}$, but no two nodes of the same layer are linked. The standard type of RBM has binary-valued neuron nodes and also bias weights. Depending on the particular task, RBM can be trained in either supervised or unsupervised ways.

43.4.2 CNN

A convolutional neural network (CNN) is initially designed to analyze visual imagery. Typically, CNN contains the following layers as shown in Fig. 43.8: the input layer, the convolutional layer, the pooling layer, the fully connected layer, and the output layer. Some CNN structures also have the normalization layer after the pooling layer. When it is used for image processing, the raw images are first input into the convolutional layer to learn the high-level and more abstract features. The convolutional layer captures the high-level latent features through multiple filters called kernels. A kernel is usually a $k \times k$ square matrix, which moves in the input image matrix from left to right and from top to bottom. A filtering operation is performed with the kernels on the corresponding positions of the input image matrix for generating high-level features. Then, the pooling layer performs a down-sampling operation on the high-level features based on the spatial dimensionality, to reduce the number of parameters. Finally, several fully connected layers are stacked to perform nonlinear transformation of the output high-level features from the pooling layers. Compared with a traditional multi-layer perceptron neural network, CNN has the following distinguishing characteristics that make it generalize well on vision problems: 3D volumes of neurons, local connectivity, and shared weights.

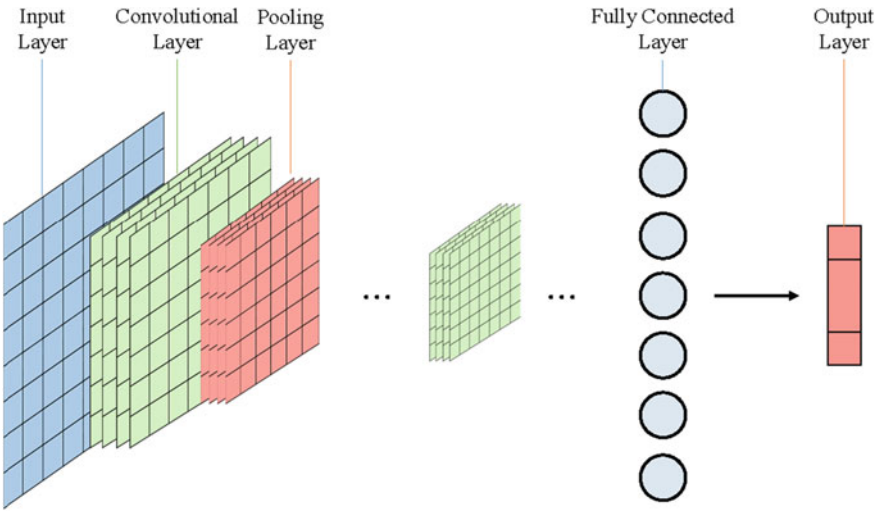


Fig. 43.8 Structure of CNN

43.4.3 RNN and LSTM

A Recurrent neural network (RNN) is designed to recognize the sequential characteristics of the input data and use the previous patterns to predict the future output. It is widely used in many areas such as speech recognition, natural-language processing, and time series data analysis. Figure 43.9 shows the general structure of an RNN network, where x_t is the input data, A are the parameters of the RNN network, and h_t is the learned hidden state. As shown in Fig. 43.9, the output of the previous time step $t - 1$ is input into the neurons of the next time step t. In this way, the historical information in the past time steps can be stored and conveyed to the future. A major shortcoming of the standard RNN is that it only has a short-term memory due to the issue of vanishing gradients. To solve this problem, the LSTM network was invented, which is capable of capturing the dependencies of the input data in a

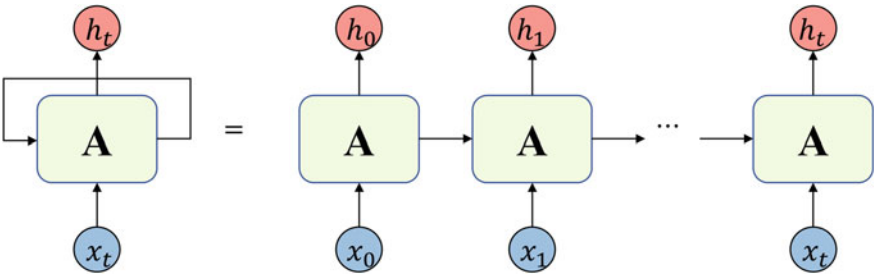


Fig. 43.9 Structure of an RNN

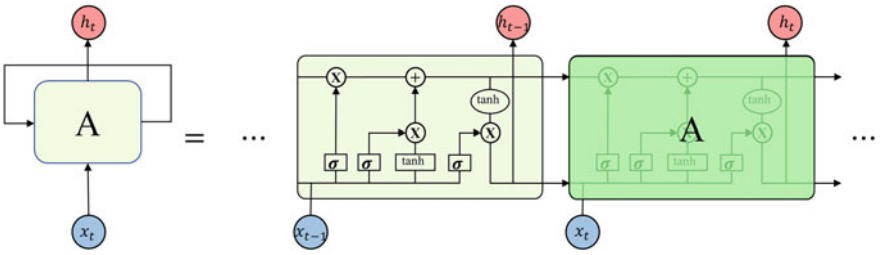


Fig. 43.10 Structure of an LSTM

much longer time period. Compared with RNN, LSTM can remember the long-term historical information of input due to its specially designed memory unit. As shown in the middle part of Fig. 43.10, an LSTM unit is composed of the following three gates: input gate, forget gate, and output gate. The input gate controls whether to let new input in, the forget gate controls whether to ignore some unimportant historical information, and the output controls whether to let the historical information impact the current output.

43.4.4 Autoencoder (AE)

An autoencoder is a type of artificial neural network that aims to learn compact data coding in an unsupervised manner (Hinton and Salakhutdinov 2013). As shown in Fig. 43.11, AE generally contains three types of layers: the input layer, the hidden layers, and the output layer. The raw data are first fed into the input layer, and then, one or multiple hidden layers are stacked to form an encoder for coding the input as compact latent representation vectors. Then, a decoder which is also composed of one or several hidden layers is used to reconstruct the raw input from the compact latent vector learned by the encoder. AE learns a compact representation of the input data in an unsupervised manner, which can be considered as a way of dimensionality reduction. As an effective learning technique for unsupervised feature representation, AE facilitates various downstream data-mining and machine-learning tasks such as classification and clustering. A stacked autoencoder (SAE) is a neural network consisting of multiple stacked AEs in which the outputs of the current AE are wired to the inputs of the successive AE (Bengio et al. 2006).

43.5 Reinforcement Learning

Reinforcement learning is more general than supervised/unsupervised learning (Richard and Andrew 1998). It learns from the interactions with the environment to get as much reward as it can over the long term. Intuitively, reinforcement learning

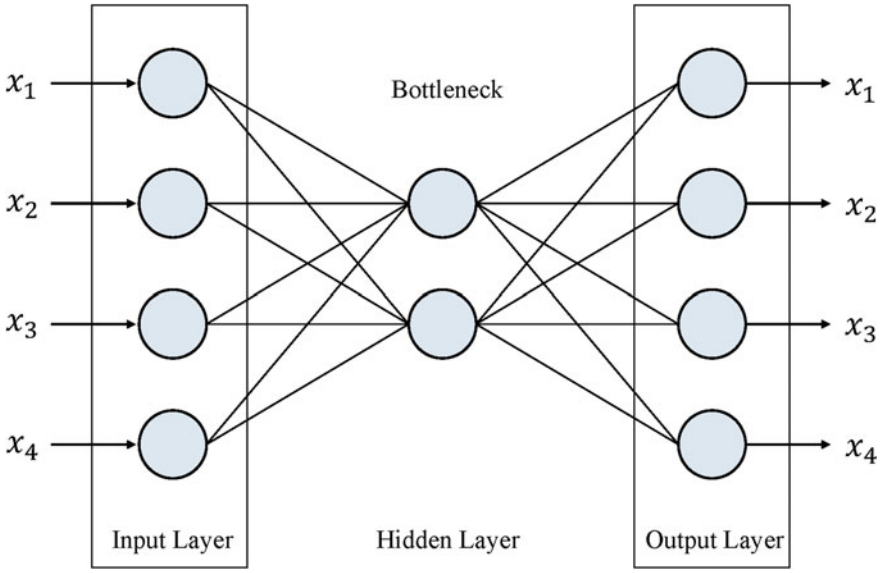


Fig. 43.11 Structure of an autoencoder

tries to imitate the human stress reaction. As shown in Fig. 43.12, imagine that you are a child in a living room with a stove in it, assume that you feel cold and are far from the stove, and then you try to approach it. You feel good and understand that the stove is a positive thing. But if you stay too close to the stove, your hand will be burned. From the interaction with the stove, you will learn that the stove is positive when you are a sufficient distance away because it produces warmth. But if you get too close to it, you will be burned. So too close to the stove will produce negative reward.

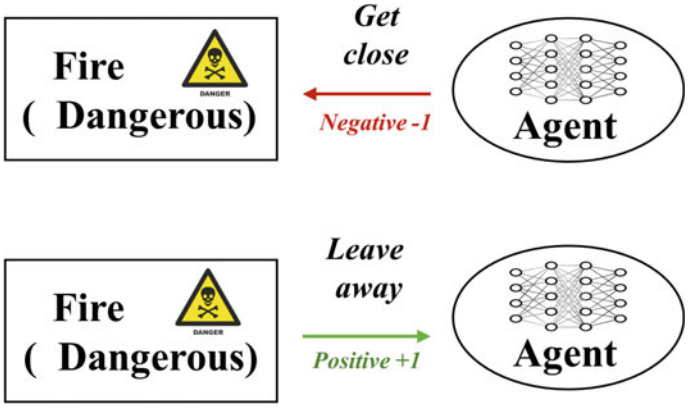


Fig. 43.12 A toy example to illustrate how humans learn through interaction with the environment

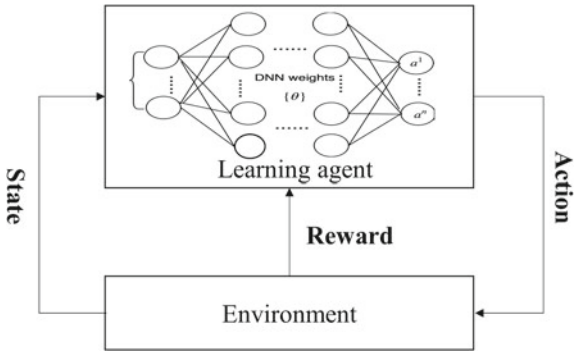


Fig. 43.13 General view of reinforcement-learning algorithms

Similar to humans learning through interaction with the environment, the reinforcement-learning algorithms learn to choose the most appropriate action through trail-and-error. The general idea of reinforcement-learning algorithms is illustrated in Fig. 43.13, which mainly consists of the four key elements: environment, reward, action, and state. A reinforcemsent-learning agent tries to learn how to best match states and actions in order to get the maximum long-term accumulated return (reward). As a result, the strategy will more frequently perform the actions that obtain positive rewards, while the actions that lead to negative punishment are less frequently performed.

Reinforcement-learning algorithms have broad application in the fields of robotics, optimal control, chess games, strategic games, flight control, missile guidance, predictive decision making, financial investment, and urban-traffic control, as they try to solve the general issues about how to best match the states and actions (Haldorai et al. 2019). Taking urban transportation as an example, where the city transportation network needs to control the traffic lights of multiple intersections and roads. Even without domain knowledge about how to control, by specifying the rule of reward, the reinforcement-learning algorithms can autonomously learn an optimal traffic light control strategy, such that all vehicles can pass the intersection in the shortest time (Rizzo et al. 2019). Even today, due to the complexity of urban-computing problems, learning control strategies through reinforcement-learning algorithms still face challenges of consuming a huge amount of computational time. However, with the development of computing power, reinforcement learning will enable an evolution from computational intelligence to artificial intelligence (Li et al. 2019).

43.6 Applications of AI Techniques in Urban Computing

The AI techniques described above have been widely applied in various urban-computing application scenarios, including urban planning, intelligent transportation systems, location-based social networks LBSNs, urban safety and security, and

urban environmental monitoring. Next, we discuss these applications in detail. For additional discussion of the use of urban-mobility data, see Chaps. 28 and 29.

43.6.1 Urban Planning

Urban planning refers to the technical and political process concerned with the design and development of land use, and especially the spaces that the public share in urban areas. The goal of urban planning is to make cities safe, healthy, and enjoyable places to live. Urban planning is a very challenging task because a lot of complex factors should be considered, such as urban-traffic flow, human mobility, POI distribution, and urban functional regions. Traditionally, urban planners need to conduct surveys to guide them in making decisions on urban planning, which is less accurate, time consuming, and labor intensive. In the big data era, a lot of data generated in the urban area are increasingly available, and such data can be used to facilitate more effective and rational urban planning. Recently, research has tried to use big data and AI techniques in various urban planning tasks such as road-network planning (Zheng et al. 2011; Berlingerio et al. 2013), functional-regions discovery; (Zheng et al. 2014a, b; Yuan et al. 2012; Manley 2014), and city-boundary detection (Ratti et al. 2010; Rinzivillo et al. 2012).

Zheng et al. (2011) used the GPS trajectories of taxicabs traveling in urban areas to detect flawed urban planning in a city. They focused on detecting the pairs of regions with salient traffic problems and discovering the linking structure as well as correlations among them. The proposed model contains two steps: city-wide traffic modeling and flawed planning detection. In citywide traffic modeling, the urban area is first partitioned into disjoint regions based on major roads, and thus each region stands for a community containing some neighborhoods. Then, the origin–destination locations of the GPS trajectories of taxicabs are mapped to the partitioned regions, so that in each hour of a day the region transition matrices can be constructed. In flawed planning detection, the skyline of each region transition matrix is first detected, and then, a graph pattern-mining method is used to identify flawed planning from the skylines. Berlingerio et al. (2013) studied how to use large-scale cellphone mobility data of users to help transit operators better perform urban transportation planning. A system called AllAboard was developed for optimizing public transport with the guidance of people's cellphone data. AllAboard first infers the origin–destination (OD) flows in the city through a large volume of people's mobile phone location data. The OD flows are then converted to ridership on the existing transit network. Next, the sequential travel patterns are extracted from the flow data over the transit network, which can be used to propose new candidate transit routes. Finally, an optimization model is proposed to evaluate which new routes would best improve the existing transit network to increase ridership.

A functional region refers to a geographic area centered around a specific focal point with a specific function such as education, business, or transportation. Automatic functional-regions discovery and identification are particularly helpful to many

urban-computing applications such as urban planning and city management. Yuan et al. (2012) proposed a data-driven approach called DRoF to discover different functional regions of a city by using both the human-mobility data among regions and the POI distributions in the regions. DRoF first segments a city into disjointed regions based on the major roads such as arterial roads, highways, and urban expressways. Then, the functions of each region are inferred by a proposed graphic-based probabilistic inference model. By borrowing the idea from topic model in natural-language processing, DRoF regards a region as a document, a region function as a topic, and the human-mobility trips (when people reach or leave which region) as words. The POI distribution in each region is also incorporated as the side information to help the model achieve more accurate inference accuracy. Evaluations are conducted on the three-month taxi GPS trajectory data generated by over 12,000 taxicabs in Beijing. Nine types of different functional regions labeled by humans are identified by DRoF. Manley (2014) applied the community-detection algorithm over the traffic network of a city to identify functional urban regions. The traffic network was constructed from the travel routes of about 1.5 million minicab trips. The region communities discovered from the large volume of traffic flow data can help identify areas of the road network that are used together, and thus help city planners to have a better understanding of the functional structure of the city. People's mobile phone data of a city can be also used to understand the spatio-temporal distribution of people in different regions of the city. For example, call detail records (CDR), which provide information on the locations of mobile phones where a call is made or a text message is sent, can be used to infer the dynamics of urban land use (Toole et al. 2012). A supervised classification algorithm is used to identify clusters of functional zones that present similar mobile phone activity patterns.

As the city expands rapidly and people move among different regions of the city, the boundaries of a city and its regions change quickly. It is very challenging for traditional methods to capture the dynamics of city boundaries. To tackle this issue, recently there have been studies using human-mobility data or activity data (e.g., GPS trajectories and CDR data) to better discover the real borders of city regions with data-driven approaches. Ratti et al. (2010) proposed a novel approach for regional delineation by analyzing networks of billions of individual human transactions. Given a geographic area and some measure of the strength of links between its inhabitants, Ratti et al. (2010) partitioned the area into disjoint smaller regions based on the rule that the disruption to each person's links in different regions should be minimized. The proposed method was tested on a large human interaction network containing 20.8 million nodes, which is inferred from a large telecommunications database in Great Britain. The human interaction network can be also inferred from other types of data such as the vehicle GPS tracks. Rinzivillo et al. (2012) first extracted region clusters from the human-interaction network constructed from the vehicle GPS data. Then, the region clusters were mapped back onto the territory of a city and were shown to match well with the existing administrative city borders.

43.6.2 Urban Transportation

Currently, most vehicles are installed with GPS devices for real-time positioning and navigation. The large-scale vehicle GPS data reflect the urban-traffic conditions in real time and thus are crucially important for intelligent transportation systems. Both deep-learning models and traditional machine-learning models are used to address various issues in urban transportation such as traffic flow prediction (Zhang et al. 2019a, b; Du et al. 2019) and traffic-congestion prediction (Wang et al. 2015; Wang et al. 2016a, b).

To address the issue that traditional traffic flow-prediction methods cannot effectively capture the nonlinear, stochastic, and time-varying characteristics of the traffic data, Zhang et al. (2019a, b) proposed a network-scale deep traffic-prediction model GCGAN. The framework of the GAGAN model is shown in Fig. 43.14, which combines adversarial training and graph CNN. GCGAN is a prediction framework

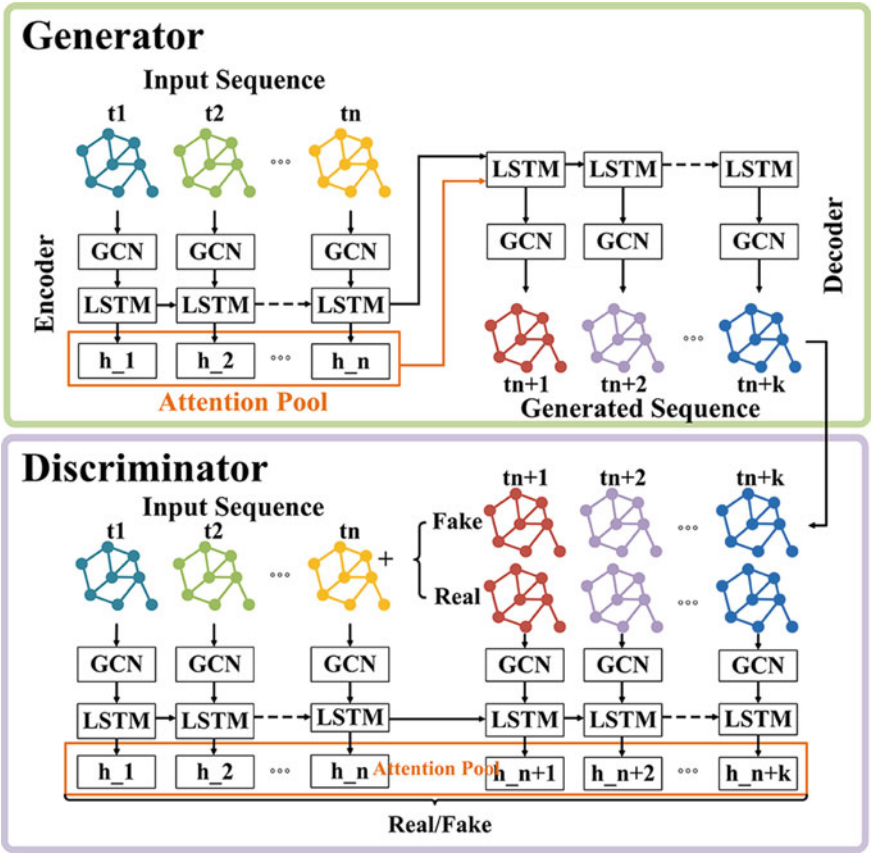


Fig. 43.14 Framework of the GCGAN model (Zhang et al. 2019a, b)

based on a Generative Adversarial Net, and thus can make more robust predictions by introducing adversarial training loss. As shown in the upper part of Fig. 43.14, GCGAN uses an encoder–decoder framework that is sequence-to-sequence based to encode the traffic conditions of a road network in previous time intervals and to decode the traffic conditions in future time intervals as the prediction. To model the spatial correlations among the road links of a transportation network, a graph convolution network (GCN) is used in both the generator and the discriminator for feature learning. LSTM is also used to capture the temporal dependencies. Du et al. (2019) studied the problem of predicting urban-traffic passenger flows with various types of traffic passenger flow data, including subway, taxi, and bus flows. Considering the complex factors such as hybrid transportation lines, mixed traffic models, transfer stations, and some extreme weather, a deep irregular convolutional residual LSTM network model called DST-ICRL was proposed by Du et al. (2019). The passenger flows among different traffic lines in a transportation network are first modeled as multi-channel matrices analogous to the RGB pixel matrices of an image. Then, a deep-learning framework that integrates an irregular convolutional residential network and LSTM units is proposed to learn the spatial–temporal feature representations from the passenger flow matrices. DST-ICRL samples both the short-term and long-term historical traffic data for model training to capture both the periodicity and the long-term trend of the traffic passenger flows.

Although deep-learning models are popular nowadays, some traditional machine-learning models such as matrix factorization and Markov models may perform better when there are multiple types of heterogeneous traffic data that need to be fused for traffic analysis. Wang et al. (2015) used a coupled matrix and tensor factorization model to infer city-wide traffic-congestion conditions by fusing multiple types of data including social-media data, social-event data, road physical features, and traffic-congestion patterns. As shown in Fig. 43.15, the proposed model used a coupled matrix and tensor factorization scheme to collaboratively factorize the traffic-congestion matrix X with the congestion correlation matrix Z, event tensor A, and the road feature matrix Y. By assuming that these matrices and tensor share the common latent factor matrix U in the road-segment dimension, these data are jointly factorized in order to fuse all the information. The traffic-congestion matrix of an entire city is then completed by multiplying the low-rank latent factor matrices U and V. Wang et al. (2016a, b) further extended the model of Wang et al. (2015) by incorporating GPS probe data. Wang et al. (2016a, b) constructed two traffic-congestion matrices: one was inferred from social-media data and the other from GPS probe data. The final estimation result is the weighted combination of the two matrices. Wang et al. (2016a, b) proposed an extended coupled hidden Markov model (E_CHMM) to combine GPS probe data and social-media data for traffic-congestion prediction. Figure 43.16 shows the framework of E_CHMM, which contains a data collection and processing part and the model part. Besides the vehicle GPS probe data, the tweets that report traffic events are also collected and used in this model. From each traffic-related tweet, the traffic event type, location, and time information are extracted. For each road link, Wang et al. (2016a, b) assumed that the occurrence

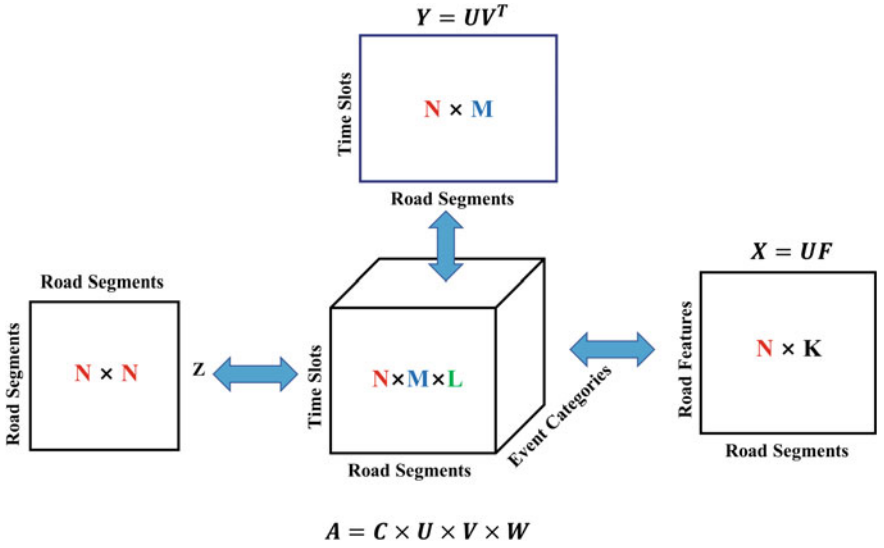


Fig. 43.15 Coupled matrix and tensor factorization model for traffic-congestion estimation (Wang et al. 2019, 2020)

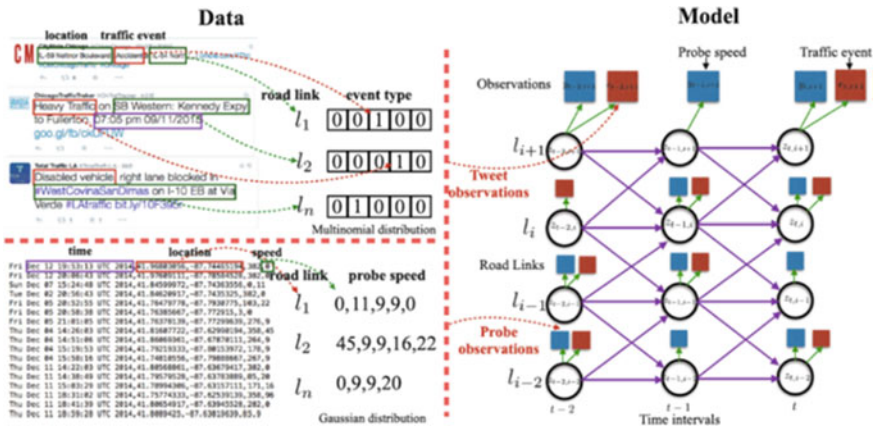


Fig. 43.16 Extended coupled hidden Markov model (E_CHMM) for traffic-congestion prediction (Wang et al. 2016a, b)

of traffic events follows a multinomial distribution, and the traveling speed of vehicles in a particular time interval follows a Gaussian distribution. In the model part, the traffic-congestion states of the road links in a road network are hidden and need to be inferred, while the GPS probe readings and traffic events extracted from tweets

are observations. The goal of E_CHMM is to accurately infer the hidden traffic-congestion states of a road network based on the fusion of two types of observations: GPS probe readings and traffic event-related tweets.

43.6.3 Location-Based Social Networks (LBSNs)

LBSNs such as Foursquare and Flickr are social networks that use GPS features to locate users and enable users to share their locations and contents to their friends through mobile devices. They are more and more popular as they can connect users in both physical and virtual worlds. When users come to favorable restaurants, new POIs, or tourist attractions, they can check-in through their mobile phones immediately, so that their friends nearby can know their locations and join. AI techniques can be used to support many applications in LBSNs, including next check-in location prediction or recommendation (Ye et al. 2010; Gao et al. 2013; Bao et al. 2012), potential friends recommendation (Scellato et al. 2011; Bao et al. 2015), and check-in time prediction (Yang et al. 2018).

In LBSNs, there usually exist strong social and geospatial ties among users and their favorite locations. To take this into consideration for better check-in location recommendation, Ye et al. (2010) proposed a novel friendly collaborative filtering (FCF) approach for location recommendation based on the collaborative ratings on the places made by social friends. Motivated by the fact that a user's preferences for the check-in locations may change continuously over time, Gao et al. (2013) considered the temporal effects in location recommendation in LBSNs. Two types of temporal properties of a user's daily check-in preferences were considered: (1) non-uniformness, which means that a user has different check-in preferences at different hours of a day; and (2) consecutiveness, which means that a user's check-in preference in consecutive hours is more similar than that in non-consecutive hours. The two properties demonstrate that a user's check-in time and the corresponding preferred check-in locations can be highly correlated. Therefore, Gao et al. (2013) proposed a new check-in location recommendation framework by considering the temporal effects based on the observed two temporal properties. Besides a user's preference, other factors such as a user's current location and the opinions about a location given by the others may also be helpful for location recommendation. Bao et al. (2012) proposed a location-based and preference-aware recommender system that recommended POIs such as restaurants and shopping malls to a user by considering the user preferences, the current location of the user, and the opinions of the POIs given by other users.

Friend recommendation is a critically important service in social networks to help users find new friends and expand their social circles. In LBSNs, the location information can help to improve the effectiveness of social-friend recommendation. The basic intuition is that a user's preference can be revealed by his or her visited locations in LBSNs. Similar location histories imply similar preferences, thus such users are more likely to become friends (Bao et al. 2015). For example, Scellato et al.

(2011) analyzed the LBSN data from Gowalla, from which they found that the link-prediction space can be largely reduced by considering the similarity of the visited locations of the users. Based on this observation, a supervised link-predication model that considers the users' visited locations was proposed by Scellato et al. (2011) to predict which users will become friends in the future. Check-in time prediction aims to predict the time when a user will check-in to a given location. Generally, check-in time prediction can be formulated as a regression problem by considering time as a continuous variable. However, directly applying a regression model may not achieve desirable performance due to the check-in data scarcity issue. To deal with this, Yang et al. (2018) formulated check-in time prediction as a survival analysis problem and proposed a recurrent-censored regression (RCR) model to address it. RCR first uses the gated recurrent units (GRUs) to learn the latent representations of historical check-ins of a user and then inputs the latent representations into a censored regression model to predict the check-in time at a given location.

43.6.4 On-Demand Service

On-demand services (e.g., Uber, Mobike, DiDi, GoGoVan, etc.) are becoming increasingly popular nowadays due to the wide use of mobile phones and the prevalence of the sharing economy. A large volume of on-demand service data is generated continuously and needs to be analyzed in real time to help the service providers meet customer needs and improve the user experience. Many challenging tasks in on-demand services, such as demand–supply prediction (Wang et al. 2019, 2020) and user behavior prediction (Wang et al. 2017a, b), require effective AI techniques.

Wang et al. (2017a, b) studied the order response-time prediction problem in on-demand logistics services. In on-demand logistics services, users can make goods delivery orders via a mobile application, and registered van drivers would respond to take these orders in a very short period of time (usually less than several minutes). Making and taking orders through such an online app installed in mobile phones is much faster than the traditional way through van calling centers, and thus makes the logistics service much more efficient. An important task to help the service providers improve their services is the accurate prediction of the response time of the van drivers to the posted delivery orders, because the response time can largely reflect the preference of the drivers for the order. Wang et al. (2017a, b) formulated the response-time prediction task as a matrix factorization problem, and proposed a coupled sparse matrix factorization model to fuse the heterogeneous and sparse data from different domains, including historical order data, personalized requirements of the user, and location-relevant features, for more accurate prediction. Currently, dockless bike-sharing systems have emerged as a new type of on-demand service in China. Users can check-out and check-in a bike conveniently at any location through scanning the QR-code on the bike with an app installed in their mobile phones. The demand–supply analysis of the bikes in dockless bike-sharing systems is a very important yet challenging problem for efficient and effective system management.

Wang et al. (2019, 2020) proposed a data-driven approach for bike usage demand–supply inference in dockless bike-sharing systems. The idea is that before massively deploying a large number of bikes in an entire city, the system operator will first pre-deploy a relatively small number of bikes in certain regions of the city for data collection. The demands in some regions are first estimated from a small number of observed bike check-out/in data directly, and then, they are used as seeds to infer bike usage demands in other regions of the city. Wang et al. (2019, 2020) formulated the problem as a matrix completion task by considering the regions and time intervals as the two dimensions of the bike usage demand and supply matrices. As the two matrices are sparse and only partial entries are known due to the bike trip data in limited regions, a matrix factorization model was designed to complete the demand and supply matrices.

Deep-learning models such as CNN and LSTM are also widely used for demand–supply prediction in on-demand services. Lin et al. (2018) proposed a graph CNN model to predict the station-level hourly demand in a large-scale bike-sharing network. The model proposed by Lin et al. (2018) combined convolutional neural networks and LSTM to learn the underlying correlations of bike usage between the bike stations. Wang et al. (2017a, b) studied the supply–demand prediction problem for online car-hailing services with deep-learning methods. An end-to-end learning framework called DeepSD was proposed by Wang et al. (2017a, b) which used a novel deep neural network structure to automatically discover complicated supply–demand patterns from the car-hailing service data.

43.6.5 Urban Safety and Security

Crimes, traffic accidents, and environmental disasters can seriously threaten urban safety and security. In the big data era, urban safety- and security-related data such as crimes and traffic accidents can be recorded and stored in a database. Recently, there has been increasing research interest in studying whether and how AI techniques can be applied to analyzing these data, and to help address various urban safety- and security-related issues such as disaster detection (Lee and Sumiya 2010; Song et al. 2013) and crime prediction (Duan et al. 2017; Huang et al. 2018).

Lee and Sumiya (2010) developed a nation-wide geo-social event detection and monitoring system by collecting a large number of messages from Twitter. The proposed geo-social event detection model contains the following main steps: (1) collecting geo-tagged tweets using a Twitter monitoring system; (2) identifying regions of interest of Twitter users and measuring geographic regularities of crowd behaviors, and (3) detecting geo-social events through a comparison of the regularities. Song et al. (2013) analyzed and modeled the evacuation behaviors of people during the Great East Japan Earthquake and Fukushima nuclear accident based on a large volume of people's real mobility data in daily life. A population mobility database was constructed to store and manage people's mobility data of GPS records

from approximately 1.6 million individuals throughout Japan over one year. A probabilistic inference model was developed to effectively represent people's mobility patterns. The proposed model can help researchers toward a better understanding of human evacuation behaviors during a disaster, and how those behaviors can be impacted by various cities during disasters. The system developed by Song et al. (2013) can be used to simulate and predict population mobility when disasters happen in cities so as to improve future disaster relief and management.

Many governments and law-enforcement agencies make city crime data (e.g., crime type, location, and time information) publicly available, so that researchers can use AI techniques for crime-data analysis. An important application of AI for crime-data analysis is crime prediction. Huang et al. (2018) developed a crime-prediction framework based on a deep neural network, called DeepCrime. DeepCrime can capture the dynamic crime patterns and explore the evolving inter-dependencies between different types of crimes to predict how many crime incidents will occur in the future in different regions of a city. A region-category interaction encoder is used to learn the complex interactions between regions and occurred crime categories. Then a hierarchical recurrent framework was proposed to jointly encode the temporal dynamics of crime patterns and capture the inherent interrelations between crimes and other ubiquitous data such as POIs. Finally, an attention mechanism was used to capture the unknown temporal relevance and automatically assign importance weights to the learned hidden states in different time frames. Duan et al. (2017) applied deep convolutional neural networks (CNNs) for automatic crime-referenced feature extraction and crime prediction. The urban area under study was first divided into grid regions. Then, the crimes in all the grid regions can be considered as an image, where each grid region is a pixel and the crime number is the gray value of the pixel. CNNs are applied on the image-like crime data of all the grid regions for feature learning.

43.6.6 Urban Environment Monitoring

Currently, a large number of diverse sensors are deployed all around a city to monitor environmental variables, weather conditions, and air-quality indexes (AQI) in real time. With a large amount of data collected from these sensors, AI techniques are required to process and analyze the data for smart environment monitoring.

Some air-quality monitoring stations have been built in different locations to collect a city's real-time air-quality indexes (AQI) such as $PM_{2.5}$, NO_2, and CO. However, due to the high cost of building and maintaining such stations, only a very limited number of stations can be built in a city; it is then a challenge to accurately obtain the AQI data of the entire city. Zheng et al. (2013) inferred the fine-grained AQI throughout a city by fusing the AQI data of limited locations with other types of data, including the meteorology, traffic flow, human mobility, structure of road networks, and POIs. A semi-supervised learning approach based on the co-training framework was proposed. This approach contains an artificial neural network to model the spatial

correlation between the AQI of different locations, and a temporal classifier to model the temporal dependency of AQI in a location. Cheng et al. (2018) proposed a deep-learning model named ADAIN for urban air-quality inference. ADAIN combines feedforward and recurrent neural networks for modeling static and sequential features as well as capturing deep feature interactions effectively. An attention mechanism was also applied in a pooling layer of ADAIN to automatically learn the different weights of features from different monitoring stations.

Due to population expansion in big cities, urban noise pollution currently is becoming a more and more serious issue that threatens public health. AI techniques can also be used to help monitor, estimate, and analyze urban noise. Rana et al. (2010) designed an end-to-end participatory urban noise mapping system called Ear-Phone. Ear-Phone leverages compressive sensing to address the issue of recovering the noise map from the incomplete and random samples obtained by crowdsourcing noise-pollution data. The noise data are collected by the sound sensors installed in mobile phones. Zheng et al. (2014a, b) studied how to infer the fine-grained noise situation, including a noise-pollution indicator and the composition of noises at different times of a day in New York City, by using multi-sourced data including citizens' complaint data about city noise, social media, road-network data, and POIs. The noise situation of New York City was modeled as 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, the noise situation throughout New York City can be recovered.

43.7 Conclusion

Recently, mining knowledge from the data generated in urban spaces for supporting urban-computing tasks to help build smart cities is a critically important and substantially challenging research topic. The large volume of heterogeneous data that are continuously generated in urban spaces, and recent advances in AI techniques, especially deep learning, have provided us with unprecedented opportunities to tackle the big challenges in urban computing. In this chapter, we conducted a comprehensive review of the challenges, methodologies, and frameworks that arise when AI techniques are applied in urban computing, and categorized the application domains of urban computing. To address the unique challenges for learning knowledge from urban data, we introduced both the traditional AI techniques and recently popular deep-learning models that are widely used for urban computing, including supervised learning, semi-supervised learning, unsupervised learning, matrix factorization, graphic models, deep learning, and reinforcement learning. We also categorized the utilization of AI techniques in different urban-computing applications including urban planning, urban transportation, location-based social networks (LBSNs), urban safety and security, and urban environmental monitoring.

References

Bao J, Zheng Y, Mokbel MF (2012) Location-based and preference-aware recommendation using sparse geo-social networking data. In: 20th international conference on advances in geographic information systems, Redondo Beach, California, 06–09 Nov

Bao J, Zheng Y, Wilkie D, Mokbel MF (2015) Recommendations in location-based social networks: a survey. GeoInformatica 19(3):525–565

Bengio Y, Lamblin P, Popovici D, Larochelle H (2006) Greedy layerwise training of deep networks. In: 19th international conference on neural information processing systems, Canada, 04–07 Dec

Berlingerio CM, Giusy F, Lorenzo GD, Nair R, Pinelli F, Sbodio ML (2013) All aboard: a system for exploring urban mobility and optimizing public transport using cellphone data. In: 12th European conference on machine learning and principles and practice of knowledge discovery in databases. Washington DC, USA, 12–16 Jan

Caruana R, Niculescu-Mizil A (2006) An empirical comparison of supervised learning algorithms. In: 23rd international conference on machine learning, Pittsburgh, Pennsylvania, USA, 25–29 June

Castro-Neto M, Jeong YS, Jeong MK, Han LD (2009) Online-SVR for short-term traffic prediction under typical and atypical traffic conditions. Expert Syst Appl 36(3):6164–6173

Cheng WY, Shen YY, Zhu YM, Huang LP (2018) A neural attention model for urban air quality inference: Learning the weights of monitoring stations. In: Thirty-second AAAI conference on artificial intelligence, New Orleans, Louisiana, USA, 02–07 Feb

Daniel DL, Sebastian SH (2000) Algorithms for non-negative matrix factorization. In: 13th international conference on neural information processing systems, Denver, CO, USA, 27–30 Nov

Du BW, Peng H, Wang SZ, Bhuiyan MZA, Wang LH, Gong Q, Liu L, Li J (2019) Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction. IEEE Trans Intell Transp Syst. https://doi.org/10.1109/TITS.2019.2900481

Duan L, Hu T, Cheng E, Zhu JF, Gao C (2017) Deep convolutional neural networks for spatiotemporal crime prediction. In: International conference on Information and Knowledge Engineering (IKE), Las Vegas, Nevada, USA, 17–20 July

Gao HJ, Tang JL, Hu X, Liu H (2013) Exploring temporal effects for location recommendation on location-based social networks. In: 7th ACM conference on recommender systems, Hong Kong, China, 12–16 Oct

Giannotti F, Nanni M, Pedreschi D, Pinelli F (2007) Trajectory pattern mining. In: 13th ACM SIGKDD conference on knowledge discovery and data mining, San Jose, California, USA, 12–15 Aug

Haldorai A, Ramu A, Murugan S (2019) Artificial intelligence and machine learning for future urban development. Springer International Publishing, New York

Hinton GE, Salakhutdinov RR (2013) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507

Huang C, Zhang J, Zheng Y, Chawla NV (2018) Deepcrime: attentive hierarchical recurrent networks for crime prediction. In: 27th ACM international conference on information and knowledge management, Torino, Italy, 22–26 Oct

Ishwarappa Anuradha A (2015) A brief introduction on big data 5Vs characteristics and Hadoop technology. Proc Comput Sci 48:318–324

Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. The MIT Press, Cambridge, Massachusetts and London, England

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444

Lee R, Sumiya K (2010) Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection. In: ACM SIGSPATIAL GIS workshop on location based social networks, San Jose, California, 02 Nov

Li GL, Randy G, Keisuke N, He B (2019) Human-centered reinforcement learning: A survey. IEEE Transac Hum Mach Syst 49(4):337–349

Lin L, He Z, Peeta S (2018) Predicting station-level hourly demand in a large-scale bike-sharing network: a graph convolutional neural network approach. Transport Res Part C: Emerg Technol 97:258–276

Manley E (2014) Identifying functional urban regions within traffic flow. Reg Stud Reg Sci 1(1):40–42

Rana RK, Chou CT, Kanhere SS, Bulusu N, Hu W (2010) Ear-phone: An end-to-end participatory urban noise mapping system. In: 9th ACM/IEEE international conference on information processing in sensor networks, Stockholm, Sweden, 12–16 April

Ratti C, Sobolevsky S, Calabrese F, Andris C, Reades J, Martino M, Claxton R, Strogatz SH (2010) Redrawing the map of Great Britain from a network of human interactions. PLoS ONE 5(12):e14248. https://doi.org/10.1371/journal.pone.0014248

Richard SS, Andrew G (1998) Barto. In: Reinforcement learning: a introduction. MIT press, Cambridge

Rinzivillo S, Mainardi S, Pezzoni F, Coscia M, Pedreschi D, Giannotti F (2012) Discovering the geographical borders of human mobility. KI - Künstliche Intelligenz 26(3):253–260

Rizzo GS, Vantini G, Chawla S (2019) Reinforcement learning with explainability for traffic signal control. In: 22nd IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October

Rosenblatt F (1957) The perceptron, a perceiving and recognizing automaton project. Cornell Aeronautical Laboratory, Buffalo, New York

Scellato S, Noulas A, Mascolo C (2011) Exploiting place features in link prediction on location-based social networks. In: 17th ACM SIGKDD international conference on knowledge discovery and data mining, San Diego, California, USA, 21–24 Aug

Shang JB, Zheng Y, Tong WZ, Chang E, Yu Y (2014) Inferring gas consumption and pollution emission of vehicles throughout a city. In: 20th SIGKDD conference on knowledge discovery and data mining, New York, New York, USA, 24–27 Aug

Song X, Zhang Q, Sekimoto Y, Horanont T, Ueyama S, Shibasaki R (2013) Modeling and probabilistic reasoning of population evacuation during large-scale disaster. In: 19th SIGKDD conference on knowledge discovery and data mining, Chicago, Illinois, USA, 11–14 Aug

Toole JL, Ulm M, Bauer D, González MC (2012) Inferring land use from mobile phone activity. In: ACM SIGKDD international workshop on urban computing, Beijing, China, 12 Aug

Wang SZ, He LF, Stenneth L, Yu PS, Li ZJ (2015) Citywide traffic congestion estimation with social media. In: 23rd ACM SIGSPATIAL international conference on advances in geographic information systems, Seattle, Washington, 03–06 Nov

Wang SZ, He LF, Stenneth L, Yu PS, Li ZJ, Huang ZQ (2016a) Estimating urban traffic congestions with multi-sourced data. In: 17th IEEE international conference on mobile data management, Porto, Portugal, 13–16 June

Wang SZ, Li FX, Stenneth L, Yu PS (2016b) Enhancing traffic congestion estimation with social media by coupled hidden Markov model. In: Joint European conference on machine learning and knowledge discovery in databases, Riva del Garda, Italy, 19–23 Sept

Wang D, Cao W, Li J, Ye JP (2017a) Deepsd: supply-demand prediction for online car-hailing services using deep neural networks. In: IEEE 33rd International Conference on Data Engineering (ICDE), San Diego, CA, USA, 19–22 April

Wang YQ, Cao JN, He LF, Li WG, Sun LC, Yu PS (2017b) Coupled sparse matrix factorization for response time prediction in logistics services. In: ACM on conference on information and knowledge management, Singapore, Singapore, 06–10 Nov

Wang SZ, Chen H, Cao JN, Zhang JW, Yu PS (2019) Locally balanced inductive matrix completion for demand-supply inference in stationless bike-sharing systems. IEEE Transac Knowl Data Eng. https://doi.org/10.1109/TKDE.2019.2922636

Wang SZ, Cao JN, Yu PS (2020) Deep learning for spatio-temporal data mining: a survey. In: IEEE transactions on knowledge and data engineering. https://doi.org/10.1109/TKDE.2020.3025580

Yang GL, Cai Y, Reddy CK (2018) Spatio-temporal check-in time prediction with recurrent neural network based survival analysis. In: Twenty-Seventh international joint conference on artificial intelligence, Stockholm, Sweden, 13–19 July

Ye M, Yin PF, Lee WC (2010) Location recommendation for location-based social networks. In: 18th SIGSPATIAL international conference on advances in geographic information systems, San Jose, California, 02–05 Nov

Yuan J, Zheng Y, Xie X (2012) Discovering regions of different functions in a city using human mobility and POIs. In: 18th SIGKDD conference on knowledge discovery and data mining, Beijing, China, 12–16 Aug

Zhang JF, Hua XS, Huang JQ, Shen X, Chen JY, Zhou Q, Fu ZH, Zhao YR (2019a) City brain: practice of large-scale artificial intelligence in the real world. IET Smart Cities 1(1):28–37

Zhang YX, Wang SZ, Chen B, Cao JN (2019) GCGAN: generative adversarial nets with graph CNN for network-scale traffic prediction. In: International joint conference on neural networks, Budapest, Hungary, 14–19 July

Zheng WC, Zheng Y, Xie X, Yang Q (2010) Collaborative location and activity recommendations with GPS history data. In: 19th international conference on World Wide Web, Raleigh, North Carolina, USA, 26–30 April

Zheng Y, Liu YC, Yuan J, Xie X (2011) Urban computing with taxicabs. In: 13th international conference on ubiquitous computing, Beijing, China, 17–21 Sep

Zheng Y, Liu F, Hsieh HP (2013) U-Air: When urban air quality inference meets Big Data. In: 19th SIGKDD conference on knowledge discovery and data mining, Chicago, Illinois, USA, 11–14 Aug

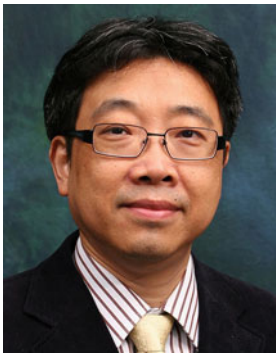
Zheng Y, Capra L, Wolfson O, Yang H (2014a) Urban computing: concepts, methodologies, and applications. ACM Trans Intell Syst Technol 5(3):1–55

Zheng Y, Liu T, Wang YL, Zhu YM, Liu YC, Chang E (2014b) Diagnosing New York City's noises with ubiquitous data. In: 16th international conference on ubiquitous computing, Seattle, Washington, 13–17 Sept

Zhou X, He J, Huang GY, Zhang YC (2015) SVD-based incremental approaches for recommender systems. J Comput Syst Sci 81(4):717–733

Zhu XJ (2005) Semi-supervised learning literature review. Technical report 1530. University of Wisconsin-Madison

Senzhang Wang is an Associate Professor of the College of Computer Science and Technology at the Nanjing University of Aeronautics and Astronautics. He is a member of ACM, AAAI, and CCF. His research interests include spatio-temporal data mining and graph data mining.



Jiannong Cao is a Chair Professor of the Department of Computing at The Hong Kong Polytechnic University, and leads the Internet and Mobile Computing Laboratory. He is an IEEE Fellow, a distinguished member of ACM, and a senior member of CCF. His research interests include distributed compu-ting, wireless sensing, mobile computing, and big data.