Front. Comput. Sci., 2014, 8(2): 316–329
DOI 10.1007/s11704-014-3258-8

RESEARCH ARTICLE

Learning to detect subway arrivals for passengers on a train

Kuifei YU (✉)[1,3], Hengshu ZHU[2], Huanhuan CAO[3], Baoxian ZHANG[1], Enhong CHEN[2], Jilei TIAN[3], Jinghai RAO[3]

1 Research Center of Ubiqutious Sensor Networks, College of Engineering & Information Technology, University of Chinese Academy of Sciences, Beijing 100049, China;
2 Laboratory of Semantic Computing and Data Mining, Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China;
3 Nokia Research Center, Nokia (China) Investment Corp. Ltd., Beijing 100176, China

Abstract The use of traditional positioning technologies, such as GPS and wireless local positioning, rely on underlying infrastructure. However, in a subway environment, such positioning systems are not available for the positioning tasks, such as the detection of the train arrivals for the passengers in the train. An alternative approach is to exploit the contextual information available in the mobile devices of subway riders to detect train arrivals. To this end, we propose to exploit multiple contextual features extracted from the mobile devices of subway riders to precisely detecting train arrivals. Following this line, we first investigate potential contextual features which may be effective to detect train arrivals according to the observations from 3D accelerometers and GSM radio. Furthermore, we propose to explore the maximum entropy (MaxEnt) model for training a train arrival detector by learning the correlation between contextual features and train arrivals. Finally, we perform extensive experiments on several real-world data sets collected from two major subway lines in the Beijing subway system. Experimental results validate both the effectiveness and efficiency of the proposed approach.

Keywords subway arrival detection, mobile users, smart cities, information storage and retrieval, experimentation

Received July 24, 2013; accepted January 10, 2014

E-mail: yukuifei@gmail.com

1 Introduction

Advances in smart mobile technologies have enabled unprecedented capabilities for mobile context sensing [2]. Indeed, to enable the next generation of intelligent and personalized context-aware services [3–7], researchers have done much work on *context recognition*, which refers to recognizing the semantic meaning of a context according to multiple contextual features extracted from the outputs of mobile sensors, such as 3D accelerometers and Bluetooth radios. For example, there exists previous works on transportation mode recognition [8–10], mobile environment recognition [11,12], and user activity recognition [13–16].

In this paper, we study the problem of detecting subway arrivals by exploiting the contextual information collected from the sensors in mobile devices. This problem is an important context recognition problem that has a wide range of potential applications. In the following, we provide a case to intuitively illustrate the application of detecting train arrivals to improve user experience of subway riders.

Example *Subway arrival reminder service*

Jack enjoys reading, catnapping, or building castles in the air on the subway train so much that he involuntarily ignores recurring subway broadcasts, and he misses his destination station from time to time. However, with a subway reminder

This is a substantially extended and revised version of our earlier work [1], which appears in the Proceedings of the 14th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'13).

application on his mobile phone enabled by subway arrival detection technology, Jack can be reminded of the arrival at his destination station.

However, it is a non trivial task to detect subway arrivals, because most traditional accurate positioning technologies, such as GPS positioning, cell tower triangulation, and WiFi local positioning, are not available in a subway environment. First, mobile devices cannot receive GPS signals in the subway system. Second, cell tower triangulation positioning [17–19] cannot directly work for detecting subway arrivals either, because the cell towers serving for subway riders are usually linearly deployed along the subway lines. This is different from the deployment of cell towers on the ground. In fact, even on the ground, the errors in cell tower triangulation positioning systems can be as high as tens of meters. Given the sparse deployment of 4G cell towers in many countries due to the lack of high frequency bands [20], the accuracy of cell tower triangulation positioning will be hard to be improve in the near future. Third, while it seems that WiFi local positioning [21–23] is a good alternative approach, there are still many subway stations where there are no WiFi access points due to the high cost of deployment and maintenance. People may argue that the subway operation companies know the accurate position of each train with Automatic Vehicle Location systems, however, up to now this real-time information is still hard to obtain by third party applications and services in many countries due to a variety of reasons, including security concerns and IPR issues [24]. Even if the real-time location of trains is available to passengers in the train, the application still needs to identify which train the user is taking to predict the arrivals. It is non trivial to specify which train a user is taking without the deployment of infrastructures. Therefore, in the light of the above discussions, a precise approach for detecting subway train arrivals is needed for effective development of reminder services in subway systems.

To this end, in this paper, we propose to exploit multiple contextual features extracted from the mobile devices of subway riders to precisely detect train arrivals. Specifically, we first study several contextual features that may be effective to detect train arrivals according to the observations from 3D accelerometer and GSM radio. The contextual features include acceleration variances, cell tower switching, and received signal strength indicator (RSSI) value. Furthermore, we propose to exploit the maximum entropy (MaxEnt) model for training a train arrival detector by learning the correlation between the identified contextual features and train arrivals. The contributions of this paper are highlighted as follows:

- To the best of our knowledge, this is the first to precisely detect subway arrivals using the output of multiple mobile sensors in mobile devices. The proposed approach can enable a wide range of potential applications and services, such as subway arrival reminder services, and thus has great value in both academia and in industry.
- To study the proposed problem, we develop a subway contextual data collection application based on the Android platform and collect several real-world data sets from two major subway lines in the Beijing subway system. We will share both the data collection system and the collected data with the academic community.
- We carry out extensive experiments on real-world data sets to evaluate the proposed subway arrival detection approach with a baseline extended from a widely used approach in activity recognition. Our experimental results demonstrate both the effectiveness and efficiency of our approach.

In Section 2, we give a detailed description of the outputs of 3D accelerometers and GSM sensors, and formally define the subway arrival detection problem. In Section 3 we present the technical details of extracting contextual features presented by 3D accelerometers and GSM data, and in Section 4 we present the process of learning to detect subway arrivals from this data. In Section 5, we report experimental results. In Section 6, we provide a brief review of related work. Finally, we conclude this paper in Section 7 and introduce future work.

2 Overview

As mentioned above, most traditional positioning technologies cannot work for subway arrival detection. Therefore, we propose to leverage the outputs of multiple mobile sensors, i.e., 3D accelerometers and GSM sensors, for detecting subway arrivals. In this section, we first present the details of the outputs of 3D accelerometers and GSM sensors, and then give the formal problem statement of subway arrival detection.

2.1 3D accelerometers

3D accelerometers, or tri-axis accelerometers are widely equipped by smart phones, digital audio players, and personal digital assistants. This kind of sensor can be used for measuring the acceleration in three different axis of the mobile device and are often used to switch the screen between landscape and portrait modes. In recent years, some researchers have started to leverage mobile 3D accelerometers for human activity recognition [13,15,25]. Intuitively, when a subway train is slowing down as it arrives at a station, the 3D accelerometer of a subway rider's mobile device will capture the deceleration of the train compared with the comparatively constant motion in subway tunnels. Figure 1 illustrates a real example of 3D accelerometer output as a subway rider passes three stations. From Fig. 1 we can see that there seem to exist some patterns when the train is arriving at a station. This motivates us to study how to leverage the outputs of 3D accelerometers for detecting subway arrivals.

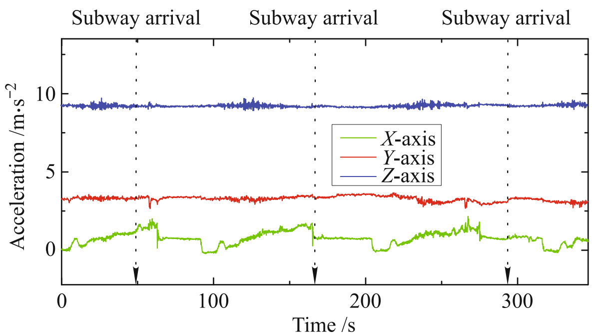

Fig. 1 A real example of 3D accelerometer readings as a subway rider passes three stations

2.2 GSM sensors

GSM sensors are standard equipment in a mobile phone and play an important role in its basic functions such as seeking a cell-site with good signal quality. It can output the signal strength and the ID of the current serving cell-site. Although this information cannot be directly used for detecting subway arrivals due to limited accuracy, it is still useful for extracting some effective features to improve subway arrival detection.

To be specific, Fig. 2 shows a real example of the shift of serving cell-sites as a subway rider passes three stations. For ease of the illustration, each cell-site is assigned a unique index from 1 to 21. From Fig. 2 we can observe that a station

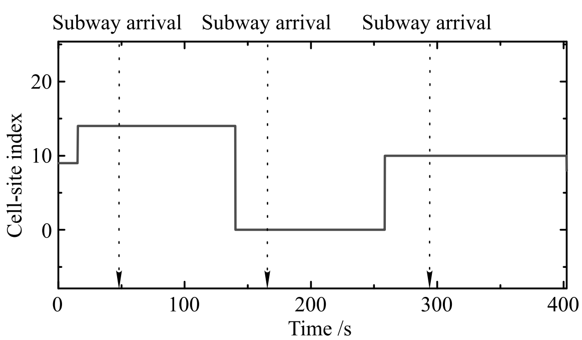

Fig. 2 A real example of the shift of serving cell-sites as a subway rider passes three stations

is usually covered by a single cell-site. Thus, a shift of serving cell-site should indicate the train is between two stations and will arrive at next station within a particular time window. It is possible to extract some effective features based on the shift of serving cell-sites for subway arrival detection.

The signal strength of the current serving cell-site may also be useful for subway arrival detection since this value is usually relatively high and stable when the subway train in a station, and drops as the subway train moves into tunnels. For example, Fig. 3 shows the signal strength as a passenger passes three stations; the signal strength is measured by RSSI [26]: a widely used metric and defined in arbitrary strength unit (ASU) ranging from 0 (weakest) to 31 (strongest) in the Android system. From Fig. 3, we observe that there exists some patterns in RSSI when the train arrives at a station and these patterns may be able to be used as effective features for indicating subway arrivals.

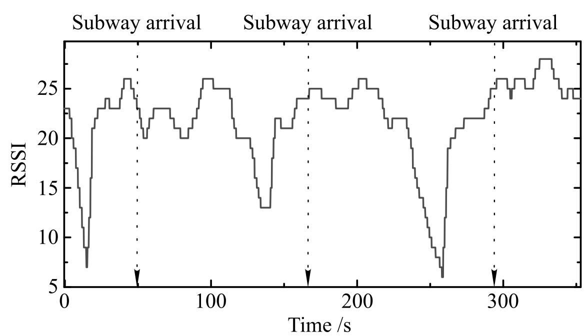

Fig. 3 A real example of the change of RSSI as a subway rider passes three stations

2.3 Problem statement

To facilitate the description of the following sections, we first define the problem of subway arrival detection as follows:

Definition Station arrival detection

Given a detection interval Δt, the objective of subway arrival detection problem is to map the latest contextual feature set $C_t = \{\langle f^k : v^k \rangle\}$ into two semantic labels $l = \{arrival, non\text{-}arrival\}$ every Δt time points, where $\langle f^k : v^k \rangle$ denotes the kth contextual feature and the corresponding value which is calculated at time point t.

It is worth noting that the values of the contextual features used for detecting subway arrivals are not real time values. They are only the latest values calculated at time point t and may be slightly different from the current values. If the calculation of these features or the detection model is not efficient enough, we may not be able to capture the event of a subway arrival in time. Therefore, in practice we should select contextual features that have low calculation complexities to reduce the delay and select a fast detection model. Moreover, we should select some contextual features which are not real time sensitive, such as the deviation of RSSI in the past few seconds when leveraging RSSI for detecting subway arrivals.

According to the definition, the subway arrival detection problem can be converted to a supervised classification problem. We divide this problem further into two parts: *how to extract effective contextual features from the raw outputs of sensors?* And, *how to train an effective station arrival detection model through machine learning technologies?* The solutions for the two sub-problems are presented in the subsequent sections.

3 Contextual feature analysis

In this section, we study several contextual features extracted from the outputs of 3D accelerometers and GSM sensors, and provide preliminary analysis of their effectiveness for detecting subway arrivals. For analyzing each contextual feature, we studied collected data over ten subway stations but only illustrate three of them for simplicity in this section.

3.1 3D accelerometer based features

As discussed in the previous section, the outputs of 3D accelerometers may be useful for subway arrival detection because their acceleration will change dramatically during the short time window when a subway train arrives at a station. Thus, in this section we investigate several features based on the outputs of 3D accelerometers that are potentially useful for detecting subway arrivals.

3.1.1 The variance of single dimension accelerations

The variance of a single dimension acceleration is calculated from the K accelerations in this dimension sampled over the past K time points. According to a previous study on responsiveness [27,28], one second is an important limit of response time, and this is the limit for uninterrupted user thought flow. Since the sample rate of 3D accelerometers is approximately one every 150 ms we can get six measurements per second (in 900 ms): we set $K = 6$.

Figure 4 shows the acceleration readings from a smart phone on the X-axis v_X, Y-axis v_Y, and Z-axis v_Z as the subway rider passes three stations where $K = 6$. From Fig. 4 we can observe that $variance(v_X)$ and $variance(v_Y)$ have obvious peaks when the train arrives at stations. By contrast, the peaks of $variance(v_Z)$ do not seem to be relevant to subway arrivals. This phenomenon results from the way a subway rider holds the smart phone. If the rider changes the orientation in which they hold the phone, $variance(v_Z)$ may become relevant to subway arrivals while $variance(v_X)$ or $variance(v_Y)$ may lose the indicating ability. This is worth noting that because the acceleration variance is real number, we divide the values of this feature into several predefined ranges in order to ensure generality. To be specific, we map the variance of one dimension accelerations into the following ranges: $[0, 0.05)$, $[0.05, 0.1)$, $[0.15, 0.2), \ldots, [0.95, 1.0)$, and $[1.0, \infty)$.

3.1.2 The mean variance of 3D accelerations

Compared to the variance of single dimension accelerations, the mean variance of three dimension accelerations may be a better contextual feature to indicate subway arrivals since it is not sensitive to the orientation in which the subway rider holds the phone. This is because no matter which axis captures the dramatic variance of acceleration of the train slowing down, the information can be reflected by the mean variance of three dimension accelerations. For example, Figure 5 shows the combined mean of $variance(v_X)$, $variance(v_Y)$, and $variance(v_Z)$ for a smart phone passing three stations. From Fig. 5 we observe that this feature can also indicate subway arrivals well. Moreover, for ensuring the generality of the feature, we map the mean variance of three dimension accelerations into the following ranges: $[0, 0.01)$, $[0.01, 0.02), \ldots,$ $[0.29, 0.30)$, and $[0.30, \infty)$.

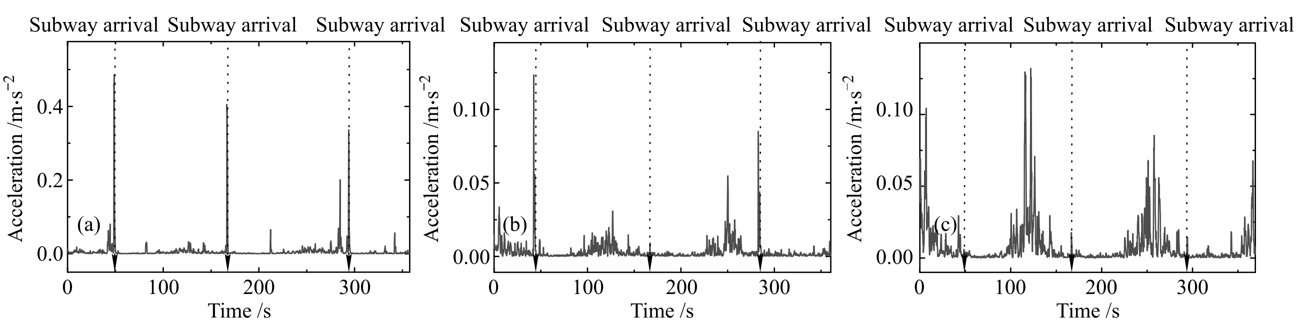

Fig. 4 The variance of a smart phone's accelerations on (a) X-axis (b) Y-axis and (c) Z-axis when the subway rider passed by three stations

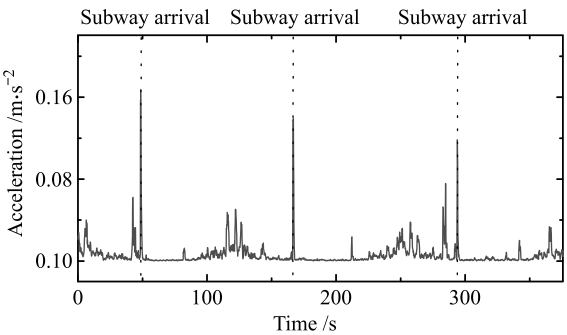

Fig. 5 The combined mean of *variance* (v_X), *variance* (v_Y), and *variance* (v_Z) for a smart phone as a subway rider passes three stations

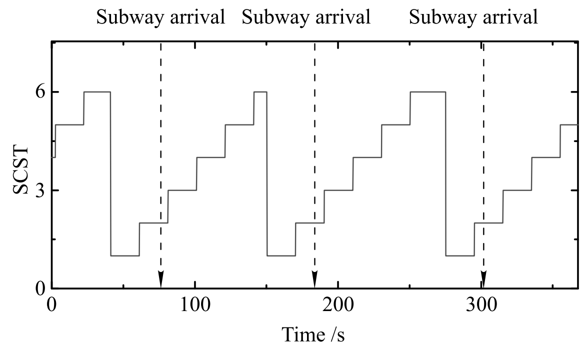

Fig. 6 The change of SCST for a smart phone passing three subway stations

3.2 GSM sensor based features

As discussed in the previous section, the output of GSM sensors, i.e., the serving cell-site and signal strength, may be able to indicate subway arrivals. In this section, we investigate the effectiveness of several GSM sensor based features for subway arrival detection.

3.2.1 The cell tower switching

As mentioned above, we observe that a shift of serving cell-site usually happens when a train is moving from one station to another. Intuitively, the subway will arrive at the next station after a certain time period once the cell-site has shifted. Therefore, we define serving cell-site shift time (SCST) to capture the relationship between subway arrivals and the shift between serving cell-sites, which can be calculated by

$$\mathrm{SCST} = t - t_{\mathrm{shift}}, \tag{1}$$

where t denotes the time when a subway arrival detection model makes a detection and t_{shift} denotes the time point of the most recent shift between serving cell-sites. Moreover, to guarantee the generality of this feature, we map the value of SCST into five classes according to the observations of the real-world data set: $Class_1$ (0 s < SCST < 20 s), $Class_2$ (20 s < SCST < 40 s), $Class_3$ (40 s < SCST < 60 s), $Class_4$ (60 s < SCST < 80 s), $Class_5$ (80 s < SCST < 100 s), and $Class_6$ (SCST > 100 s). Figure 6 shows the changes of SCST for a smart phone passing three subway stations. From Fig. 6 we observe that when the train arrives at a station, the corresponding SCST is always mapped into $Class_2$, which validates the effectiveness of SCST.

3.2.2 The signal strength of the serving cell-site

Intuitively, when a train remains in a station, the corresponding signal strength is stable and relatively high. Therefore, we can directly leverage the RSSI value of signal strength as a contextual feature for detecting subway arrivals. To guarantee the robustness of this feature, we map the RSSI value into four levels according to the standards of the Android system—*Very Strong*: 3 (RSSI $\in [24, 31]$), *Strong*: 2 (RSSI $\in [16, 23]$), *Weak*: 1 (RSSI $\in [8, 15]$), and *Very Weak*: 0 (RSSI $\in [0, 7]$).

However, leveraging RSSI values alone will introduce much noise because the signal of a serving cell-site may also be strong at some time when the train is in a tunnel. Therefore, to solve this problem, we introduce a second feature called mean absolute deviation of RSSI value (MADRV) as follows:

$$\mathrm{MADRV} = \frac{\sum_{i=1}^{N}(|V_i - \overline{V}|)}{N}, \tag{2}$$

where V_i indicates the ith sampled RSSI value before detection time t and $\overline{V}$ indicates the average value of the total N sampled RSSI values. Using MADRV values, we can measure the stability of the signal strength during the past time window. The larger MADRV, the greater the likelihood that the train will arrive at the next station soon. Figure 7 shows the change of MADRV for a smart phone when the train passes three subway stations. The received RSSI values over the past five seconds before detection time are considered. From Fig. 7 we can observe that when the train arrives

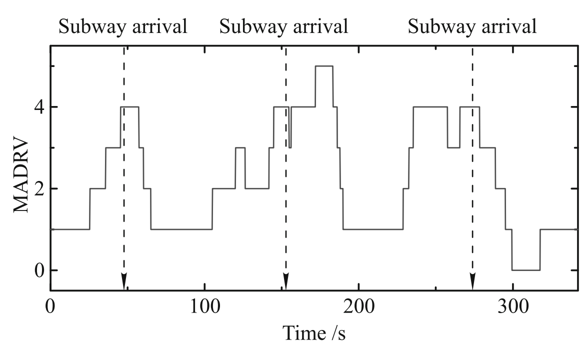

Fig. 7 The change of MADRV for a smart phone as the train passes three subway stations

at a station, the MADRV is always equal to 4, which validates the effectiveness of the feature.

The features based on 3D accelerometers and GSM are designed to be complementary to each other, and work together for detecting arrival events. For example, the mean variance of 3D accelerations is complementary to the variance of single dimension accelerations, and the RSSI feature and the time of cell ID shifts reflect different aspects of contexts on a subway train.

4 Learning to detect subway arrivals

After contextual feature extraction, the remaining work is to build a train detection model $\mathcal{M}$ that can integrate multiple effective contextual features for detecting subway arrivals. Many supervised classification models can be applied to solving this problem. In this paper, we propose to leverage the MaxEnt classifier for training a detection model for three main reasons [29,30]: 1) MaxEnt is robust and has been successfully applied to a wide range of classification applications, such as POS tagging in natural language processing, and it is proven to perform better than other alternative models in classifying imbalanced and sparse data; 2) compared with other classification approaches, MaxEnt is more flexible at incorporating different types of features, such as the various features extracted from 3D accelerometers and GSM sensors; 3) MaxEnt is very efficient in the processes of both training and testing, and this is particularly suitable for deployment on mobile devices.

4.1 MaxEnt based detection model

In our problem, given a detection time window Δt and the current time point t, MaxEnt defines the conditional probability of a subway arrival label l (i.e., whether the subway train arrives at a stop or not at t) as

$$P(l|C_t) = \frac{1}{Z(t)} \exp(\sum_i \lambda_i f_i(C_t, l)), \tag{3}$$

where C_t denotes the contextual feature set extracted at the time t, $f_i(C_t, l)$ denotes a feature function on C_t and l, λ_i indicates the weight of $f_i(C_t, l)$, and $Z(t)$ indicates a normalization factor:

$$Z(t) = \sum_l \exp(\sum_i \lambda_i f_i(C_t, l)). \tag{4}$$

The two most important problems of applying the MaxEnt model are the selection of feature functions and the learning of parameters Λ. We select the feature functions for subway arrival detection as follows. First, for each feature we discussed in Section 3, we count all of its values that appear in the training data. Then, given a feature f^k and the set of its all appearing values V^k, we generate $|V^k| \times L$ feature functions such that

$$f_{(v^k, l' : v^k \in V^k, l' \in L)}(C_t, l) = \begin{cases} 1, & (f^k : v^k) \in C_t \text{ and } l = l', \\ 0, & \text{else}, \end{cases}$$

where v^k denotes an appearing value of the feature f^k and L indicates the predefined label set.

Given a group of feature functions and a labeled training data set $\mathcal{D} = \{\langle C_t, l \rangle\}$, the objective of training a MaxEnt model is to find a set of parameters $\Lambda = \{\lambda_i\}$ that maximize the conditional log-likelihood:

$$L(\Lambda|\mathcal{D}) = \log \prod_{\langle l, C_t \rangle \in \mathcal{D}} P_\Lambda(l|C_t). \tag{5}$$

We can leverage many machine learning algorithms to train MaxEnt models, such as improved iterative scaling (IIS) [31] and limited-memory BFGS (L-BFGS) [32]. In this paper, following the comparison results of algorithms for MaxEnt parameter estimation in [32], we leverage the most efficient algorithm L-BFGS for model training. Once the parameters Λ have been learned using a training data set, we infer the label l^* according to a contextual feature set $C_{t'}$ by

$$l^* = \arg\max_l P(l|C_{t'}, \Lambda). \tag{6}$$

4.2 Imbalanced classification problem

When we take subway arrival detection as a supervised classification problem, a critical challenge is that training samples with the label *arrival* are extremely limited compared with others. To be specific, in our data sets the average ratio of label *arrival* to *non-arrival* is only 0.006 3. If we use such imbalanced data to train a detection model, the classification accuracy of the subway arrival would be very poor. This problem is known as the *imbalanced classification problem* and has been well-studied by many researchers [33–35]. Although MaxEnt is a good model at dealing with imbalanced training data, it still suffers the extreme imbalance of our data set. To solve this problem, we propose to leverage two widely used approaches to imbalanced classification: data under-sampling and data over-sampling [33,35,36]. Data under-sampling aims to reduce the number of training samples with the label *non-arrival* (*negative samples*), and data over-sampling aims to duplicate the training samples with the label *arrival* (*positive samples*). To the best of our knowledge, how to select the best value of drop rate or duplicate rate is still an open question. Therefore, in this paper we do not give any principles but compare various settings to evaluate the detection accuracy in experiments.

5 Experiments

In this section, we evaluate our approach through extensive experiments on several real-world data sets collected from two major subway lines in Beijing.

5.1 Data collection and preprocessing

To study the problem of detecting subway train arrivals through mobile devices, we developed a subway context data collection application on Android device for collecting the context data that are potentially useful for detecting subway arrivals, including the outputs of 3D accelerometers and GSM sensors[1]. In order to collect the 3D accelerometer data and GSM data we designed three SQLite tables. Manually observed data is recorded in subway_data_ls, 3D accelerometer data in subway_data_acc, and GSM data in subway_data_cid. The main fields of these tables and examples are listed in Tables 1–3.

Table 1 The main fields of subway_data_ls and examples

Line_name	Station_name	Operation	Collect_time
Line#10	Xitucheng	LEAVE	1326675655759
Line#5	Ciqikou	STOP	1326676026056
Line#10	Zhichunlu	LEAVE	1328060958451

Table 2 The main fields of subway_data_acc and examples

Acc_x	Acc_y	Acc_z	Collect_time
–0.632 069	2.489 97	9.327 81	1328060735389
0.536 301	2.777 27	9.251 19	1328060789656
1.283 29	3.122 04	9.078 81	1328060808957

Table 3 The main fields of subway_data_cid and examples

CellID	Signal_strength	Collect_time
19725	26	1328061658370
19739	23	1328061836604
19739	21	1328061857360

The application is developed for Google Android 2.3. The sample rate of each sensor is set according to the APIs provided by the Android system [37]: *SensorManager.SENSOR_DELAY_NORMAL* corresponding to approximately six readings per second. The sample rates of both the current serving cell-site and RSSI from GSM sensors are set to 100 ms. Moreover, the application has a user friendly interface (as illustrated in Fig. 8) for subway riders to manually record the time of train arrival or departure: this data is used as the ground truth in our experiments.

Fig. 8 The user interface of the developed subway context data collection application

To prepare the experimental data, we installed the subway context data collection application on an HTC Z710e smart phone and collected many context data from two major subway lines in Beijing. Five data sets were collected from line 10 and one from line 5 [38]. Figure 9 shows a map of subway lines in Beijing [38]. The details of the collected data sets are listed in Table 4.

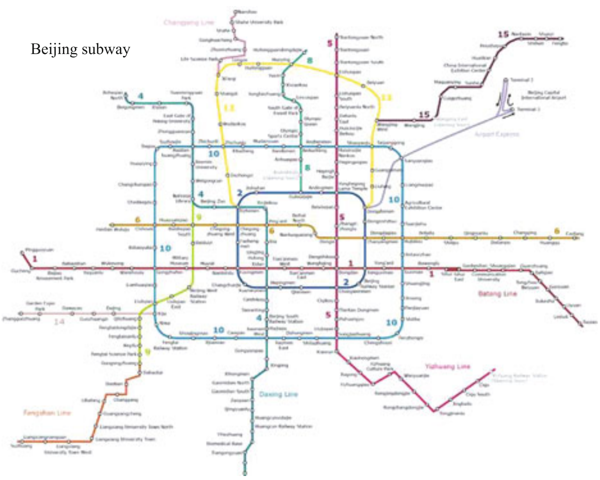

Fig. 9 The map of Beijing subway lines, where line 5 and line 10 are selected for data collection

Table 4 The details of collected subway context data

Data set	Number of 3D acc. records	Number of GSM records	Number of subway arrivals
Line 10–1	13 153	15 892	18
Line 10–2	6 460	8 752	10
Line 10–3	13 097	16 302	18
Line 10–4	13 516	16 439	18
Line 10–5	7 711	9 984	11
Line 5	9 889	11 594	14

To extract training and test samples from the collected con-

[1] This data set will be made publicly available on May 1st, 2014 at http://home.ustc.edu.cn/žhs/dataset.htm.

text data, we first determine the detection interval Δt as the interval of 3D accelerometer outputs. Then, for each time t when a 3D accelerometer output is recorded, we build a sample by extracting the contextual features mentioned in above section. For the variance of 3D accelerations, we consider the most recent six 3D accelerometer outputs (around one second). For GSM based features, we use the most recent sampled serving cell-site and RSSI as the current results at t and consider the most recent 30 samples (around five seconds) when calculating the MADRV. Finally, we assign a sample with time stamp t and arrival label *arrival* if a real arrival happened at t. Otherwise, the sample will be assigned the label *non-arrival*. Considering the delay of confirming subway arrivals when collecting subway context data, if the interval between t and a marked subway arrival time is less than 0.5 s, we will regard that a subway arrival happened at t.

5.2 Benchmark method and evaluation metrics

To the best of our knowledge, there is no existing method of subway arrival detection that can be directly used as a benchmark method to evaluate our approach. Therefore, in this paper we extend a widely used approach in transportation mode recognition as the baseline [8,9]. Specifically, we calculate the mean of $variance(v_X)$, $variance(v_Y)$, and $variance(v_Z)$ for every six 3D accelerometer outputs. If the mean value is larger than a predefined threshold ϕ, we label this detection time as *arrival*, where ϕ is set to be the minimum mean variance of 3D accelerations for all positive samples in the training data.

In order to study the contribution of 3D accelerometer and GSM based features separately, we also evaluate two MaxEnt models with only one kind of feature, namely, ME-3D (MaxEnt with 3D accelerometers based features) and ME-G (MaxEnt with GSM sensor based features). We also explore a combined MaxEnt model that combines all the contextual features discussed here (denoted as ME-3D-G). To avoid over fitting in the training process of the MaxEnt model, we use the Gaussian prior for parameter Λ similar to [39]. All the above approaches are implemented in C++ and the experiments are conducted on a PC with a 3GHz×4 quad-core CPU, with 3G RAM.

To evaluate the performance of subway arrival detection, we use *Recall*, *Precision*, and F_{score} with respect to the *arrival* label for measuring the outputs of each test approach. However, although this traditional method reflects the ability of detection models to capture the relevance of arrival label to corresponding features, they do not directly reflect the performance of subway arrival detection from a user experience perspective. In practice, a reasonable time delay, say two or three seconds, in subway arrival detection will not impact user experience too much. Therefore, we also propose some user experience ($_{UX}$) based metrics with more tolerance for the detections as follows.

$$Recall_{UX} = \frac{\#Hit\_Session}{\#Arr\_Session}, \tag{7}$$

$$Precision_{UX} = \frac{\#Hit\_Session}{\#Hit\_Session + \#Error}, \tag{8}$$

$$F_{UX} = 2 \times \frac{Recall_{UX} \times Precision_{UX}}{Recall_{UX} + Precision_{UX}}, \tag{9}$$

where *Arr_Session* denotes a time window which contains the $\pm N$ seconds of a sample labeled *arrival* as the ground truth, *Hit_Session* denotes a *Arr_Session* that contains at least one sample labeled *arrival* by the detection model, and *Error* denotes a sample that is labeled *arrival* by the detection model but does not fall into any *Arr_Session*. Compared to traditional *Recall*, *Precision* and F_{score}, these less strict metrics can be used for evaluating the performance of subway arrival detection with respect to user experience.

Here we set N to 5 based on our riding experience on subways. We notice that the subway broadcasting service reminds subway passengers to prepare for train arrival about 5 seconds before arrival, and passengers in the train usually take several seconds to get off clearly after train arrives. We select ±5 seconds based on these preliminary observations on the subway.

5.3 Strategy for reducing data imbalance

As mentioned previously, to solve the problem of imbalanced classification, we should select a proper drop rate for under-sampling training and a proper duplicate rate for over-sampling training. To this end, we first study the impact of different drop rates and duplicate rates on subway arrival detection performance. In our experiments, a five-fold cross validation is conducted. We first randomly divide the samples from the all six data sets into five equal parts, and then use each part as test data while using the other four parts as training data in five test rounds. Finally, we report the average performance of each approach in the five rounds of tests.

Figure 10 shows the ME-3D-G detection performance of our approach, with respect to different drop rates in the under-sampling training, where the training samples with the label *non-arrival* are randomly dropped under each drop rate. From Fig. 10 we can observe that both *Precision* (Fig. 10(a)) and $Precision_{UX}$ (Fig. 10(b)) first fall slightly as drop rate increase and then fall sharply over a certain drop rate, whereas both *Recall* and $Recall_{UX}$ roughly increase with an increase of drop rate. Moreover, both F_{score} and F_{UX} first increase and then drop with the increase of drop rate.

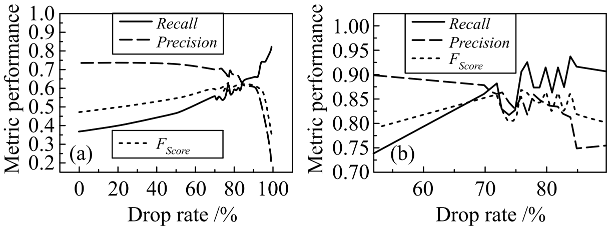

Fig. 10 The detection performance of our approach with respect to different drop rates. (a) Classic; (b) UX

Figure 11 shows the detection performance of our approach with respect to different duplicate rates in the over-sampling training. From Fig. 11 we can observe that both *Precision* (Fig. 11(a)) and $Precision_{UX}$ (Fig. 11(b)) consistently fall with the increase of duplicate rate while both *Recall* and $Recall_{UX}$ consistently rise until reaching an optima with the increase of duplicate rate. Both F_{score} and F_{UX} first increase and then become to drop with the increase of duplicate rate at a small duplicate rate.

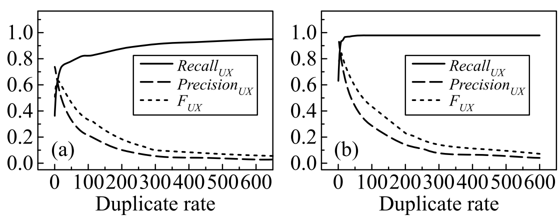

Fig. 11 The detection performance of our approach with respect to different duplicate rates. (a) Classic; (b) UX

The above experiments show that the detection performance of our approach can be improved dramatically by selecting a proper drop rate for the under-sampling approach and a proper duplicate rate for the over-sampling approach implying that the impact of sample imbalance to the detection performance is significant. Although the selection of drop or duplicate rates are still open questions, we empirically select them according to the above experimental results for building a working detection model. In the following experiments of ME-3D-G, we set the drop rate to be 76% for under-sampling and duplicate rate to be 7 for over-sampling.

Similarly, we also study the impact of different drop rates and duplicate rates on the detection performance of another two MaxEnt models with different types of features: ME-3D and ME-G. The curves of each approach are similar to those of ME-3D-G illustrated in Figs. 10 and 11, though the optimal settings of drop rate and duplicate rate for each model are different. For the sake of brevity we omit those figures here and in the following experiments, we set the drop rates for ME-3D, ME-G to be 72% and 99%, respectively, and set the duplicate rates for ME-3D, ME-G to be 5 and 55, respectively.

It is worth noting that the baseline method is not impacted by either the drop or duplicate rate; it is a naive method that only relies on unique positive samples, a change in the number of negative samples will not influence its performance. Similarly, duplicating positive samples does not increase unique positive samples and thus will equally not influence its performance.

5.4 The reusability of the model

Given a subway arrival detection model trained on the context data from a subway line, the reusability of the model indicates its ability to be applied to other context data from the same subway line. In other words, if a detection model has good reusability, once it is trained on context data collected from one subway line, we can state it can be directly used by all this line's subway riders. In order to study the reusability of our detection models and the baseline method, we first leverage a five-fold cross validation to evaluate the detection performance of each approach for the five data sets collected from line 10. We select each data set as the test data and other four as the training data for five times and report the average performance.

Table 5 shows the average detection performance of each approach in the five-fold cross validation. For each MaxEnt model, both the results with under-sampling and over-sampling are illustrated. And, for each metric, the method with the best performance is highlighted. From Table 5 we can observe the detection performance of ME-3D-G dramatically outperforms the baseline with respect to most metrics (four out of all six metrics) no matter whether we adopt the under-sampling or over-sampling approach. Although the baseline slightly outperforms ME-3D-G with respect to *Recall* and $Recall_{UX}$, its *Precision* and $Precision_{UX}$ are too low (i.e., 0.139 6 and 0.123 8) to be applied for real applications.

We also observe that ME-3D is comparable to the baseline, although it under-performs ME-3D-G with respect to each metric. To be specific, it dramatically outperforms the baseline with respect to *Precision* and $Precision_{UX}$ but dramatically under-performs the latter with respect to *Recall* and $Recall_{UX}$. If we comprehensively consider all above metrics,

Table 5 The average detection performance of each approach in the five-fold cross validation for the data sets collected from line 10

		Recall	*Precision*	F_{score}	$Recall_{UX}$	$Precision_{UX}$	F_{UX}
ME-G	Under-sampling	0.596 7	0.013 6	0.026 5	0.964 7	0.028 1	0.054 5
	Over-sampling	0.239 4	0.011 8	0.022 5	0.635 4	0.042 5	0.078 4
ME-3D	Under-sampling	0.477 6	0.265 2	0.337 0	0.769 9	0.343 9	0.469 9
	Over-sampling	0.528 6	0.245 2	0.333 9	0.837 9	0.311 4	0.452 9
ME-3D-G	Under-sampling	0.579 3	0.692 6	0.618 2	0.908 5	0.861 1	0.874 0
	Over-sampling	0.626 2	0.642 5	0.622 4	0.942 5	0.798 0	0.856 4
Baseline		0.846 9	0.139 6	0.235 0	1.000 0	0.123 8	0.215 3

F_{score} and F_{UX} can be used since they are designed for this purpose. Along this line, we can state in this experiment that ME-3D outperforms the baseline because it achieves better F_{score} and F_{UX}. It validates the motivation of leveraging the 3D accelerometer output through an elegant model over a naive rule based method, that is, although the *Recall* falls, the overall performance of arrival detection is indeed improved.

Finally, the performance of ME-G is relatively poor in the main due to its very low *Precision* and $Precision_{UX}$; this implies that GSM based features are not suitable to be used as sole contextual features for detecting subway arrivals. However, they can dramatically improve the performance of a model which only relies on 3D accelerometer based features; this is demonstrated by the advantages of ME-3D-G over ME-3D.

5.5 The extensibility of the model

A practical problem in the real applications of subway arrival detection is that it is too expensive to collect the context data from all subway lines in a country. Therefore, it is desirable for a detection model trained on context data collected from one subway to be adaptable to the context of another subway line. To this end, we evaluate the extensibility of our detection model, that is, the ability to apply it to a new subway line whose context data do not appear in the training data. We first train detection models from the data sets collected from line 10. Then we use the data set collected from line 5 as a test data to evaluate the detection performance.

Table 6 illustrates the experimental results of each approach. Compared with the experimental results illustrated in Table 5, we see that the performance of both ME-3D-G and the baseline fall with respect to most metrics, but ME-3D-G still dramatically outperforms the baseline with respect to all metrics except *Recall* and $Recall_{UX}$. Moreover, the performance of ME-3D falls with respect to the basic metrics but improves with respect to the user experience based metrics. Finally, the performance of ME-G improves in most metrics though it still performs worst among all test approaches.

5.6 The efficiency of the model

As a mobile application, the efficiency of a subway arrival detection system is a critical factor for its practical application. Since the training of a detection model can be conducted in a server, we mainly concern the detection efficiency and memory cost of the detection model. Indeed, all MaxEnt based detection models discussed above are very efficient since the inference process of MaxEnt is very simple and all used contextual features are easy to extract. In our experiments, the average time for detecting an arrival in ME-3D-G, ME-3D, and ME-G are 20.1 us, 18.2 us, and 17.4 us, respectively. The memory costs of these MaxEnt based detection models are 4.57 M, 4.57 M, and 4.56 M, respectively: this is very modest given the high memory capacity of most modern smart phones.

6 Related work

Generally, related work can be grouped into two categories.

Table 6 The detection performance of each approach trained on the data sets collected from line 10 and tested on the data set collected from line 5

		Recall	*Precision*	F_{score}	$Recall_{UX}$	$Precision_{UX}$	F_{UX}
ME-G	Under-sampling	0.460 3	0.012 3	0.024 0	1.000 0	0.034 3	0.066 3
	Over-sampling	0.460 3	0.018 7	0.035 9	0.923 0	0.071 9	0.133 4
ME-3D	Under-sampling	0.269 8	0.161 9	0.202 4	0.846 2	0.333 3	0.478 2
	Over-sampling	0.317 5	0.172 4	0.223 5	0.846 2	0.323 5	0.468 1
ME-3D-G	Under-sampling	0.269 8	0.531 2	0.357 8	0.538 5	0.875 0	0.666 7
	Over-sampling	0.301 6	0.558 8	0.391 8	0.615 4	0.888 9	0.727 3
Baseline		0.841 3	0.099 4	0.177 8	1.000 0	0.074 7	0.139 0

The first category is context recognition, which focuses on recognizing the semantic meaning of a context according to multiple contextual features extracted from the outputs of multiple mobile sensors, such as 3D accelerometers, Bluetooth sensors, and GPS. For example, Reddy et al. [8,9] studied the problem of creating a transportation mode, e.g., driving, walking, or biking, classification system that runs on a mobile phone equipped with both a GPS receiver and a 3D accelerometer. They trained the classifier using a decision tree followed by a discrete hidden markov model (DHMM) based on a real-world data set collected from six individuals. Mayrhofer et al. [12] proposed an architecture to recognize and predict mobile user context by leveraging multiple heterogeneous sensors, such as WLAN, GSM, and Bluetooth radios. They studied several user behavior based contextual features and combined them into a classification model trained by hidden markov models (HMMs). Mantyjarvi et al. [11] studied how to leverage collaborative filtering approach to recognize context for mobile users with hand-held devices. Their proposed approach can collaboratively recognize the current context of a group of users with hand-held devices, thus the recognition accuracy can be improved with respect to the approaches that only leverage the data from single devices. User activities, such as walking, sitting and standing, also provide very important contextual information. Brezmes et al. [13] studied how to recognize user activities from 3D accelerometer data on a mobile phone. They proposed a real time classification system for some basic human movements. Similarly, Kwapisz et al. [15] studied the same problem and built a data collection system based on an Android system to collect sensor data from twenty-nine mobile users. To ease the workload of annotating a large number of sensor readings for the activity recognition problem, Hu and Yang [14] proposed a transfer learning based framework for activity recognition via sensor mapping. To the best of our knowledge, although there is much related work on context recognition, detecting train arrivals for subway passengers on the train is still under-explored.

The second category is accurate positioning technologies. Among these technologies, GPS positioning is the most widely used but suffers relatively high energy consumption. Therefore, many researchers have studied other novel positioning technologies. For example, Deblauwe and Ruppel [17] proposed a solution for energy efficient positioning by combing both GPS and GSM cell ID information in mobile devices. Similarly, Paek et al. [19] proposed a cell ID aided positing system (CAPS) that can leverage a continuous cell ID sequence and the position history of smart phone users to achieve energy efficient positioning. Another drawback of GPS positioning is that it cannot be used in an enclosed space due the lack of GPS signals. To that end, Liu et al. [21] investigated several advanced indoor positioning techniques and systems, including cell-site based triangle positioning and WiFi positioning. Although these approaches can achieve good positioning performance in some cases, they cannot be leveraged for detecting subway arrivals due to the different deployment pattern of cell-sites along subway lines and the lack of WiFi access points. To provide transit tracking and arrival time prediction services, EasyTracker [40] utilized GPS embedded in smart-phones to determine routes served, locate stops, and infer schedules. Although this aimed to provide services to passengers, the solution does not work underground as it depends on GPS. Moreover, EasyTracker provides a countdown clock service to the users on bus stations, not aboard a train. Therefore, in this paper we proposed a novel subway arrival detection approach based on the outputs of both 3D accelerometers and GSM sensors to detect subway arrivals for the mobile users on the train.

7 Concluding remarks

In this paper, we studied the problem of detecting subway arrivals for passengers on a train to enable subway arrival reminding services. The key idea of our approach is to collectively combine the evidence from multiple contextual sources that are collected by various sensors in mobile devices. Specifically, we provide several effective contextual features extracted from 3D accelerometers and GSM sensors for detecting subway arrivals according to the observations of real-world use cases. Then, we combined all the extracted contextual features into a widely used machine learning model named MaxEnt for training an effective detecting model. In addition, we developed a subway contextual data collection system to collect several real-world data sets from two major subway lines in Beijing. Finally, the experimental results on these real-world data demonstrated both the effectiveness and robustness of our approach.

As illustrated in our experiments, for a subway arrival detection model, extendability is critical for its success in practical applications. A training data set which reflects more common properties of contextual data from all subway lines and fewer special properties of the contextual data from some particular subway lines may help to improve the extendability of a detection model. Random user behavior, such as giving way to others, will affect the effectiveness of 3D accelerometer based features and thus affect the detection accuracy. In future work, we plan to investigate some effective methods of selecting such common subway contextual data, and collect more kinds of noisy data to develop new detection models.

Acknowledgements We thank the anonymous reviewers for their insightful comments and discussions. We would also like to give special thanks to Prof. Hui Xiong for his valuable comments on this work. This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61173158, 61101133).

References

1. Yu K, Zhu H, Cao H, Zhang B, Chen E, Tian J, Rao J. Learning to detect the subway station arrival for mobile users. In: Proceedings of the 14th Intelligent Data Engineering and Automated Learning. 2013, 49–57
2. Han L, Jyri S, Ma J, Yu K. Research on context-aware mobile computing. In: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications Workshops. 2008, 24–30
3. Yu C C, Chang H P. Towards context-aware recommendation for personalized mobile travel planning. In: Proceedings of the 2013 Context-Aware Systems and Applications. 2013, 121–130
4. Huang C M, Lin S Y, Hsieh T H. The personalized context-aware mobile advertisement system using a novel approaching detection method over cellular networks. Software: Practice and Experience, DOI: 10.1002/spe.2208, 2013
5. Yu K, Zhang B, Zhu H, Cao H, Tian J. Towards personalized context aware recommendation by mining context logs through topic models. In: Proceedings of the 1st Pacific Asia Knowledge Discovery and Data Mining. 2012, 431–443
6. Zhu H, Chen E, Yu K, Cao H, Xiong H, Tian J. Mining personal context-aware preferences for mobile users. In: Zaki M J, Siebes A, Yu J X, Goethals B, Webb G I, Wu X, eds. Proceedings of the IEEE International Conference on Data Mining. 2012, 1212–1217
7. Zhu H, Cao H, Chen E, Xiong H. Mobile App Classification with Enriched Contextual Information. IEEE Transactions on Mobile Computing, Pre-print, DOI: 10.1109/TMC, 2013, 113
8. Reddy S, Burke J, Estrin D, Hansen M, Srivastava M. Determining transportation mode on mobile phones. In: Proceeding of the 12th IEEE International Symposium on Wearable Computers. 2008, 25–28
9. Reddy S, Mun M, Burke J, Estrin D, Hansen M, Srivastava M. Using mobile phones to determine transportation modes. ACM Transactions on Sensor Networks, 2010, 6(2): 13
10. Zhang L, Qiang M, Yang G. Mobility transportation mode detection based on trajectory segment? Journal of Computational Information Systems, 2013, 9(8): 3279–3286
11. Mantyjarvi J, Himberg J, Huuskonen P. Collaborative context recognition for handheld devices. In: Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications. 2003, 161–168
12. Mayrhofer R, Radi H, Ferscha A. Recognizing and predicting context by learning from user behavior. In: Proceedings of the 2003 International Conference on Advances in Mobile Multimedia. 2003, 25–35
13. Brezmes T, Gorricho J L, Cotrina J. Activity recognition from accelerometer data on a mobile phone. Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living. Berlin: Springer, 2009, 796–799
14. Hu D H, Yang Q. Transfer learning for activity recognition via sensor mapping. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence. 2011, 3: 1962–1967
15. Kwapisz J R, Weiss G M, Moore S A. Activity recognition using cell phone accelerometers. ACM SIGKDD Explorations Newsletter, 2011, 12(2): 74–82
16. Thiemjarus S. A device-orientation independent method for activity recognition. In: Proceedings of the 2010 International Conference on Body Sensor Networks. 2010, 19–23
17. Deblauwe N, Ruppel P. Combining gps and gsm cell-id positioning for proactive location-based services. In: Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Networking&Services. 2007, 1–7
18. Yang J, Varshavsky A, Liu H, Chen Y, Gruteser M. Accuracy characterization of cell tower localization. In: Proceedings of the 12th ACM International Conference on Ubiquitous Computing. 2010, 223–226
19. Paek J, Kim K H, Singh J P, Govindan R. Energy-efficient positioning for smartphones using cell-ID sequence matching. In: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. 2011, 293–306
20. 4G—Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/4G. 2013. Online; accessed on 22-June-2013
21. Liu H, Darabi H, Banerjee P, Liu J. Survey of wireless indoor positioning techniques and systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2007, 37(6): 1067–1080
22. Lee J Y, Yoon C H, Park H, So J. Analysis of location estimation algorithms for wifi fingerprint-based indoor localization. American Society of Transportation and Logistics, 2013, 19: 89–92
23. Arai K, Tolle H. Color radiomap interpolation for efficient fingerprint wifi-based indoor location estimation. International Journal of Advanced Research in Artificial Intelligence, 2013, 2(3): 10–15
24. Rojas F M. Transit Transparency: Effective Disclosure through Open Data. Transparency Policy Project, 2012
25. Zheng Y, Wong W K, Guan X, Trost S. Physical activity recognition from accelerometer data using a multi-scale ensemble method. In: Proceedings of the 25th Innovative Applications of Artificial Intelligence Conference. 2013, 1575–1581
26. Received signal strength indication—Wikipedia, the free encyclopedia. http://en. wikipedia.org/wiki/Received_signal_strength_indication. Online; accessed on 22-June-2013
27. Responsiveness—Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/ Responsiveness. Online; accessed on 18-Nov-2013
28. Nielsen J. Response times: the 3 important limits. http://www.nngroup.com/articles/response-times-3-important-limits/. Online; accessed on 18-Nov-2013
29. Phan X H, Nguyen L M, Horiguchi S. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web. 2008, 91–100
30. Zhu H, Cao H, Chen E, Xiong H, Tian J. Exploiting enriched contex-

tual information for mobile app classification. In: Proceedings of the 21st ACM International Conference on Information and Knowledge Management. 2012, 1617–1621

31. Della Pietra S, Della Pietra V, Lafferty J. Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(4): 380–393
32. Malouf R, Groningen R. A comparison of algorithms for maximum entropy parameter estimation. In: Proceedings of the 6th Conference on Natural Language Learning. 2002, 49–55
33. Maloof M A. Learning when data sets are imbalanced and when costs are unequal and unknown. In: Proceedings of the 2003 International Conference on Machine Learning Workshop on Learning from Imbalanced Data Sets II. 2003, 1–8
34. Sun Y, Kamel M S, Wong A K, Wang Y. Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition, 2007, 40(12): 3358–3378
35. Weiss G M. Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 7–19
36. Chawla N V. Data mining for imbalanced datasets: an overview. Data Mining and Knowledge Discovery Handbook. Berlin: Springer, 2010, 875–886
37. Android APIs. http://developer.android.com/reference/packages.html. Online; accessed on 22-June-2013
38. Beijing Subway—Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/ Beijing_Subway. Online; accessed on 22-June-2013
39. Nigam K, Lafferty J, McCallum A. Using maximum entropy for text classification. In: Proceedings of the 16th International Joint Conference on Artificial Intelligence Workshop on Machine Learning for Information Filtering. 1999, 61–67
40. Biagioni J, Gerlich T, Merrifield T, Eriksson J. Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems. 2011, 68–81

Kuifei Yu is a director of Invention Research and Development in Zhigu.com. He is a PhD candidate in the University of Chinese Academy of Sciences, Beijing, China. He received his BS from North China University of Water Resources and Electric Power, China in 2004, and his MS from Beijing University of Posts and Telecommunications, China in 2007, both in Computer Science and Technology. He joined Nokia Research Center as a researcher, then senior researcher during 2007 and 2012. His current research interests include mobile data mining, computational advertising, and patent mining. In these fields, he has applied around 30 invention patents world wide and published several papers in refereed conference proceedings and journals, such as PAKDD, ICDM, and TIST.

Hengshu Zhu is a PhD student in the School of Computer Science and Technology at the University of Science and Technology of China, Hefei. He was supported by the China Scholarship Council as a visiting research student at Rutgers—the State University of New Jersey, NJ, USA. He received his BE in Computer Science in 2009 from USTC. His main research interests include mobile data mining, intelligent data analysis, and social networks. He has published several papers in refereed conference proceedings and journals, such as CIKM, ICDM, and WWW Journal. He has been a journal reviewer for TSMC-B, WWW Journal and KAIS, and an external reviewer for several international conferences.

Huanhuan Cao is currently the principal scientist of Nuomi.com. Before joining Nuomi, he was a key research engineer of Hipu Info. Tech. Ltd, a start-up focusing on personalized news recommendation. Previously he worked for Nokia Research Center as a senior researcher. He received his BE and PhD from the University of Science and Technology of China, in 2005 and 2009, respectively. He won the Microsoft Fellow and Chinese Academic Science President Award. His current major research interests include recommender systems, location mining, and mobile user behavior analysis. In these fields, he has applied for more than 30 invention patents and published more than 20 papers in high rated conferences and journals.

Baoxian Zhang is a Full Professor of the University of Chinese Academy of Sciences, Beijing, China. He received his BS, MS, PhD degrees in Electrical Engineering from Northern Jiaotong University (Now Beijing Jiaotong University), China in 1994, 1997, and 2000, respectively. He has served as Guest Editors of special issues for ACM Mobile Networks and Applications and IEEE Journal on Selected Areas in Communications. He has served and continued to serve on technical program committees for many international con-

ferences and symposia, including IEEE GLOBECOM, ICC, and WCNC. He has published over 100 refereed technical papers in archival journals and conference proceedings. His research interests cover network architecture, protocol and algorithm design, wireless ad hoc and sensor networks, and performance evaluation. He is a senior member of the IEEE.

Enhong Chen is Professor and Vice Dean of the School of Computer Science at University of Science and Technology of China. He received his PhD in Computer Science from the University of Science and Technology of China in 1996. His general area of research is web search and data mining, with a focus on developing effective and efficient data analysis techniques for emerging data intensive applications. He has published around 100 papers in refereed journals and conference proceedings. He is a senior member of the IEEE, and a member of the ACM.

Jilei Tian is a research leader in Nokia. He received his BS and MS in Biomedical Engineering from Xi'an Jiaotong University, China and his PhD in Computer Science from the University of Eastern Finland in 1985, 1988, and 1997, respectively. He was a faculty member of Beijing Jiaotong University, China from 1988 to 1994. He joined Nokia Research Center as senior researcher in 1997, primarily in the area of spoken language processing and recently in rich context data modeling and personalized services. He has authored more than 100 publications including book chapters, journals, and conference papers. He has also around 100 patents including those pending. He has served as a member of technical committees and on the editorial board of international conferences and journals.

Jinghai Rao is a senior staff engineer at Samsung Information Systems America. Before joining Samsung, he worked in Nokia, AOL, and Carnegie Mellon University, USA. He has more than 10 years research experience in ontology, semantic web, query expansion, context awareness and location-based services. Dr. Rao is active in the artificial intelligence and knowledge representation research community. He has authored more than 30 scientific publications, and served as a program committee member or reviewer for various international conferences and journals. Dr. Rao received his PhD from Norwegian University of Science and Technology, Norway, and his MS and BS from Renmin University of China.