Learning to Detect the Subway Station Arrival for Mobile Users

Kuifei Yu[1], Hengshu Zhu[2], Huanhuan Cao[3],
Baoxian Zhang[1], Enhong Chen[2], Jilei Tian[3], and Jinghai Rao[3]

[1] University of Chinese Academy of Sciences, Beijing, China
yukuifei@gmail.com, bxzhang@ucas.ac.cn
[2] University of Science and Technology of China, Hefei, China
zhs@mail.ustc.edu.cn, cheneh@ustc.edu.cn
[3] Nokia Research Center, Beijing, China
{happia.cao,jilei.tian,jinghai.rao}@nokia.com

Abstract. The use of traditional positioning technologies relies on the underlying infrastructures. However, for the subway environment, such positioning systems may not be available for the positioning tasks, such as the detection of the train arrivals for the passengers in the train. An alternative way is to exploit the contextual information available in the mobile devices of subway riders. To this end, in this paper, we propose to exploit multiple contextual features extracted from the mobile devices of subway riders for precisely detecting train arrivals. Along this line, we first investigate potential contextual features which may be effective to detect train arrivals according to the observations from sensors. Furthermore, we propose to explore the maximum entropy model for training a train arrival detector by learning the correlations between the contextual features and the events of train arrivals. Finally, we perform extensive experiments on several real-world data sets. Experimental results clearly validate both the effectiveness and efficiency of the proposed approach.

Keywords: Subway Arrival Detection, Mobile Users, Smart Cities.

1 Introduction

Advances in smart mobile technologies have enabled unprecedented capabilities for context sensing. In this paper, we study the problem of detecting subway arrivals for the passengers in the train by exploiting the contextual information collected from the sensors in mobile devices. This problem is also an important context recognition problem which has a wide range of potential applications. In the following, we provide a case to intuitively illustrate the application of detecting train arrivals for improving the user experiences for subway riders.

Example 1 (Subway Arrival Reminding 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 had missed his destination station from time to time. However, with a subway reminding application on his

H. Yin et al. (Eds.): IDEAL 2013, LNCS 8206, pp. 49–57, 2013.

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

However, it is a nontrivial task to detect the subway arrivals, because most traditional accurate positioning technologies, such as GPS positioning, cell tower triangulation, and Wifi local positioning, are not available in the subway environment. To be specific, first, it is well known that mobile devices cannot receive GPS signals in the subway system. Second, cell tower triangulation positioning [2,7] 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. Actually, even on the ground, the errors of 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 [12], the accuracy of cell tower triangulation positioning will be difficult to be improved in the near future. Third, while it seems that Wifi local positioning [4] is a good alternative approach, there are still many subway stations where there are no Wifi accessing points. Finally, people may argue that the subway operation companies know the accurate position of each train. However, up to now, this real-time information is still difficult to be obtained by the third party applications and services in many countries due to various reasons, such as the security concerns. Therefore, in light of the above discussions, a precise approach for detecting subway train arrivals is crucially needed for the effective development of reminding services in the subway systems.

1.1 Problem Statement

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

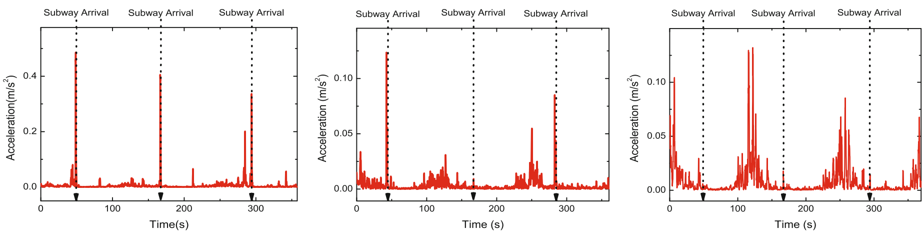
Definition 1 (Station Arrival Detection). Given a detection interval Δt, the objective of subway arrival detection problem is to map the latest contextual feature set $C_t = \{< f^k : v^k >\}$ into two semantic labels $l = \{Arrival, Non\text{-}arrival\}$ every Δt time points, where $< f^k : v^k >$ denotes the *k-th* contextual feature and the corresponding value that is calculated at time point t.

According to the definition, the subway arrival detection problem can be converted to a supervised classification problem. Therefore, this problem is divided into two parts, namely *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 following sections in detail, respectively.

Indeed, there are some related works have been reported in literatures, such as context recognition [3,8,9] and accurate positioning technologies [2,4,7]. However, to the best of our knowledge, how to detect train arrivals for subway riders is still under-explored. Therefore, in this paper we propose a novel subway arrival detection approach based on the outputs of both 3D accelerometers and GSM sensors to detecting subway arrivals.

2 Contextual Feature Analysis

In this section, we study several contextual features extracted from the outputs of 3D Accelerometers and GSM sensors and preliminarily analyze their effectiveness for detecting subway arrivals.



(a) X-axis (b) Y-axis (c) Z-axis

Fig. 1. 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

2.1 3D Accelerometer Based Features

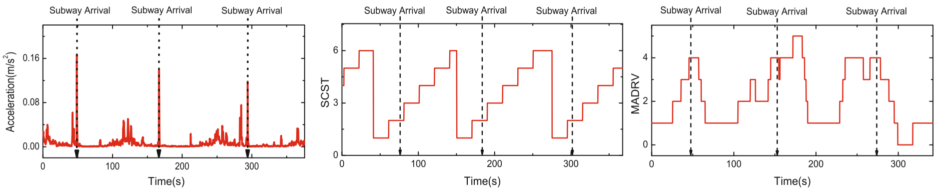
The Variance of Single Dimension Accelerations. Figure 1 shows the variances of a smart phone's accelerations on the X-axis v_X, Y-axis v_Y, and Z-axis v_Z when the subway rider passed by three stations. From this figure we can observe that $Variance(v_X)$ and $Variance(v_Y)$ have obvious peaks when the train arrives at stations. By contrast, the peaks of $Variance(v_Z)$ do not seem to be relevant to subway arrivals. Actually, this phenomenon is resulted from the way where the subway rider holds the smart phone.

The Mean Variance of Three Dimension 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 way the subway rider holds the phone. For example, Figure 2 (a) shows the mean value of $variance(v_X)$, $variance(v_Y)$, and $variance(v_Z)$ for a subway rider's smart phone when the train passed by three stations. The figure shows that this feature can indicate subway arrivals well.

2.2 GSM Sensor Based Features

The Shift of Serving Cell-Sites. We define a *Serving Cell-site Shift Time (SCST)* to capture the relevance between subway arrivals and the shift of serving cell-sites, which can be calculated by

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



(a) (b) (c)

Fig. 2. The change of (a) the mean value of $variance(v_X)$, $variance(v_Y)$ and $variance(v_Z)$, (b) SCST, and (c) MADRV for a subway rider's smart phone when the train passed by three subway stations

where t denotes the time point when a subway arrival detection model makes a detection and t_{shift} denotes the time point of the latest shift of serving cell-sites. Moreover, to guarantee the generality of this feature, we map the value of $SCST$ into five classes according to the observation of real-world data set. Figure 2 (b) shows the changes of SCST for a subway rider's smart phone when the train passed by three subway stations. From this figure we can observe that when the train arrives at a station, the corresponding SCST is always mapped into $Class_2$, which validates the effectiveness of $SCST$.

The Signal Strength of Serving Cell-Site. Intuitively, when a train stays at 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 4 levels according to the standard of Android system.

We introduce another feature called *Mean Absolute Deviation of RSSI Value* (MADRV) as follows,

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

where V_i indicates the i-th sampled RSSI value before the detection time point t and $\overline{V}$ indicates the average value of the total N sampled RSSI values. The larger MADRV, the more likelihood that the train will arrive at next station soon. Figure 2 (c) shows the change of MADRV for a subway rider's smart phone when the train passed by three subway stations. From this figure we can observe that when the train arrives at a station, the MADRV is always equal to the same value, which validates the effectiveness of the feature.

3 Learning to Detect Subway Arrivals

After contextual feature extraction, the remaining work is to train a detection model **M**, which can integrate multiple effective contextual features for detecting subway arrivals. Actually, for this problem, given a set of training samples, a lot of supervised classification models can be applied. In this paper, we propose to use the maximum entropy (MaxEnt) classifier for training a detection model.

To be specific, 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 point t, $f_i(C_t, l)$ denotes a feature function about C_t and l, λ_i indicates the weight of $f_i(C_t, l)$, and $Z(t)$ indicates a normalization factor which equals to

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

The objective of model training is to learn a set of proper parameters using training data set to maximize the model likelihood. After that we can infer the label l^* according to a contextual feature set $C_{t'}$ as $l^* = \arg\max_l P(l|C_{t'}, \Lambda)$. According to the comparison of algorithms for maximum entropy parameter estimation in [6], we use the most efficient algorithm L-BFGS for model training.

Table 1. The details of collected subway context data

Data set	#3D Acc. Records	#GSM Records	#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

3.1 Imbalanced Classification Problem

When we take subway arrival detection as a supervised classification problem, a critical challenge along the line is that the training samples with the label *Arrival* are extremely limited compared with the others. To be specific, in our data sets the average ratio of label *Arrival* to *Non-arrival* is only 0.0063. If we use such imbalanced data to train detection model, the classification accuracy of subway arrival would be very poor. This problem is known as *imbalanced classification problem*, which is well-studied by many researchers [5, 10, 11]. Although MaxEnt is a good model at dealing with imbalanced training data, it still suffers the extreme imbalance of our data sets, as illustrated in our experiments. To solve this problem, in this paper we propose to leverage the two widely used approaches to imbalance classification, namely, data under-sampling and data over-sampling [5, 11]. 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.

4 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.

4.1 Data Collection and Preprocessing

To study the problem of detecting subway train arrivals through mobile devices, we developed a subway context data collection application for collecting the context data which are potentially useful for detecting subway arrivals, including the outputs of 3D accelerometers and GSM sensors [1].

The application is developed for Google Android 2.3 System. The sampling rate of each sensor is set according to the APIs provided by Google Android system [1]. To prepare the experimental data, we installed the subway context data collection application on a HTC Z710e smart phone and collected many context data from two major subway lines in Beijing, China. To be specific, five data sets were collected from line 10 and one data set was collected from line 5 [13]. The details of the collected data sets are listed in Table 1.

To extract training and test samples from the collected context data, we first determine the detection interval, i.e., Δt, as the interval of 3D accelerometer outputs. Then, for each time point t when a 3D accelerometer output is recorded, we build a sample by extracting the contextual features .

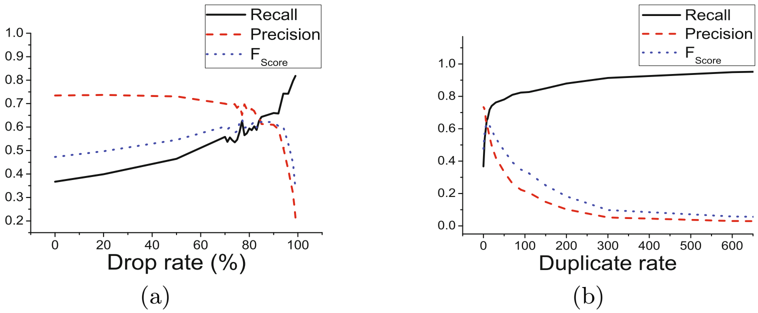


Fig. 3. The detection performance of our approach with respect to different (a) drop and (b) duplicate rates

[1] This data set will be made publicly available soon.

4.2 Benchmark Method and Evaluation Metrics

In this paper we extend a widely used approach in transportation mode recognition [8,9] as the baseline. To be specific, 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.

Moreover, to study the contribution of 3D accelerometers based features and GSM based features separately, we also evaluate two MaxEnt models with only one kind of features, namely, ME-3D (*MaxEnt with 3D accelerometers based features*) and ME-G (*MaxEnt with GSM sensor based features*), besides the MaxEnt model which combines all contextual features discussed in this paper (denoted as ME-3D-G). All above approaches are implemented by standard C++ and the experiments are conducted on a 3GHZ×4 quad-core CPU, 3G main memory PC.

To evaluate the performance of subway arrival detection, we first use $Recall$, $Precision$, and F_{score} with respect to the $Arrival$ label for measuring the outputs of each test approach. We also propose some user experience based metrics with more tolerance for the detections named $Recall_{UX}$, $Precision_{UX}$, and F_{UX}, which are correspondingly defined as $\frac{\#Hit\_Session}{\#Arr\_Session}$, $\frac{\#Hit\_Session}{\#Hit\_Session+\#Error}$, and $2 \times \frac{Recall_{UX} \times Precision_{UX}}{Recall_{UX}+Precision_{UX}}$, where $Arr\_Session$ denotes a time window which contains the ± 5 seconds of a sample labeled $Arrival$ as the ground truth, $Hit\_Session$ denotes a $Arr\_Session$ which contains at least a sample labeled $Arrival$ by the detection model, and $Error$ denotes a sample which is labeled $Arrival$ by the detection model but does not fall into any $Arr\_Session$.

4.3 The Impact of the Strategy for Reducing Data Imbalance

To solve the problem of imbalanced classification, a standard five-fold cross validation is conducted. Figure 3 (a) shows the detection performance of our approach, i.e., ME-3D-G, 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 this figure we can observe that $Precision$ first slightly drop with the increase of drop rate and sharply drop under a certain drop rate, while $Recall$ roughly increase with the increase of drop rate.

Figure 3 (b) shows the detection performance of our approach with respect to different duplicate rates in the over-sampling training. From this figure we can observe that $Precision$ consistently drop with the increase of duplicate rate while $Recall$ consistently increase until reach an optima with the increase of duplicate rate. Specially, in the following experiments of ME-3D-G, we set the drop rate to be 76% for the under-sampling approach duplicate rate to be 7 for the over-sampling approach.

Similarly, we also study the impacts of different drop rates and duplicate rates to the detection performance of another two MaxEnt Models with different types of features, i.e., ME-3D and ME-G. 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.

4.4 The Reusability of the Model

Given a subway arrival detection model trained on the context data from a certain subway line, the reusability of the model means its ability of being applied to other context data from the same subway line. In order to study the reusability of our detection models and the baseline method, we leverage a five-fold cross validation to evaluate the detection performance of each approach for five data sets collected from line 10.

Table 2. 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	0.5967	0.0136	0.0265	0.9647	0.0281	0.0545
	Over-Sampling	0.2394	0.0118	0.0225	0.6354	0.0425	0.0784
ME-3D	Under-Sampling	0.4776	0.2652	0.3370	0.7699	0.3439	0.4699
	Over-Sampling	0.5286	0.2452	0.3339	0.8379	0.3114	0.4529
ME-3D-G	Under-Sampling	0.5793	**0.6926**	0.6182	0.9085	**0.8611**	**0.8740**
	Over-Sampling	0.6262	0.6425	**0.6224**	0.9425	0.7980	0.8564
Baseline		**0.8469**	0.1396	0.2350	**1.0000**	0.1238	0.2153

Table 2 shows the average detection performance of each approach in the five-fold cross validation. From this table 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 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., 0.1396 and 0.1238) to be applied for real applications.

4.5 The Efficiency of the Model

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. To be specific, in our experiments, the average time of making a detection for ME-3D-G, ME-3D, and ME-G are 20.1us, 18.2us, and 17.4us, respectively. Moreover, the memory costs of these MaxEnt based detection models are 4.57M, 4.57M, and 4.56M, respectively, which are much less than the memory limit of most modern smart phones.

5 Concluding Remarks

In this paper, we studied the problem of detecting subway arrivals for passengers in the train, which can enable a wide range of potential applications and services, such as subway arrival reminding services. The key idea of our approach is to collectively combine the evidences from multiple contextual information, which is collected by various sensors in mobile devices.

As illustrated in our experiments, for a subway arrival detection model, the extendability is critical for its success in practical applications. A training data set which reflects more common properties of the contextual data from all subway lines and less special properties of the contextual data from some particular subway lines may help to improve the extendability of a detection model. As for our future work, we plan to investigate some effective methods of selecting such "common" subway contextual data.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China Grant Nos. 61101133 and 61173158.

References

1. Android APIs (2013), `http://developer.android.com/reference/packages.html` (accessed June 22, 2013)
2. Deblauwe, N., Ruppel, P.: Combining gps and gsm cell-id positioning for proactive location-based services. In: MobiQuitous 2007 (2007)
3. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. SIGKDD Explor. Newsl
4. Liu, H., Darabi, H., Banerijee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews
5. Maloof, M.A.: Learning when data sets are imbalanced and when costs are unequal and unknown. In: ICML 2003 Workshop on Learning from Imbalanced Data Sets II (2003)
6. Malouf, R.: A comparison of algorithms for maximum entropy parameter estimation. In: COLING 2002 (2002)
7. Paek, J., Kim, K.-H., Singh, J.P., Govindan, R.: Energy-efficient positioning for smartphones using cell-id sequence matching. In: MobiSys 2011 (2011)
8. Reddy, S., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Determining transportation mode on mobile phones. In: ISWC 2008 (2008)
9. Reddy, S., Mun, M., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Using mobile phones to determine transportation modes. ACM Trans. Sen. Netw.
10. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn.
11. Weiss, G.M.: Mining with rarity: a unifying framework. SIGKDD Explor. Newsl
12. Wikipedia. 4G — Wikipedia, the free encyclopedia (2013), `http://en.wikipedia.org/wiki/4G` (accessed June 22, 2013)
13. Wikipedia. Beijing Subway — Wikipedia, the free encyclopedia (2013), `http://en.wikipedia.org/wiki/Beijing_Subway` (accessed June 22, 2013)