

Mobile Sensing Data for Urban Mobility Analysis: A Case Study in Preprocessing

Indrė Žliobaitė, Jaakko Hollmén
Helsinki Institute for Information Technology HIIT
Aalto University School of Science, Department of Information and
Computer Science, PO Box 15400, FI-00076 Aalto, Espoo, Finland
indre.zliobaite@aalto.fi, jaakko.hollmen@aalto.fi

ABSTRACT

Pervasiveness of mobile phones and the fact that the phones have sensors make them ideal as personal sensors. Smart phones are equipped with a wide range of motion, location and environment sensors, that allow us to analyze, model and predict mobility in urban areas. Raw sensory data is being collected as time-stamped sequences of records, and this data needs to be preprocessed and aggregated before any predictive modeling can be done. This paper presents a case study in preprocessing such data, collected by one person over six months period. Our goal with this exploratory pilot study is to discuss data aggregation challenges from machine learning point of view, and identify relevant directions for future research in preprocessing mobile sensing data for human mobility analysis.

Categories and Subject Descriptors

H.2.8 [Database Applications]: Data Mining; I.5.2 [Pattern Recognition]: Feature evaluation and selection

Keywords

mobile sensing, data preprocessing, feature extraction, accelerometer, smart cities

1. INTRODUCTION

The availability and penetration of smart mobile devices is increasing; smartphone penetration in Europe is already more than 49% [2]. Mobile sensing systems are finding their way in many application areas, such as monitoring human behavior, social interactions, commerce, health monitoring, traffic monitoring, and environmental monitoring [9].

Pervasiveness of mobile phones and the fact that they are equipped with many sensor modalities makes them ideal sensing devices. Since the mobile phones are personal devices, we can use the idea of mobile sensing to probe the owner of the phone and the environment, in which the user is moving. Our general interest is to use mobile phones to

learn about the mobility patterns of people and to reason and predict about their mobility patterns in urban traffic environment.

The idea of using mobile phones as sensors is not new: mobile phones have been used for context recognition (e.g. [8]) and for measuring social interactions (e.g. [4]) in complex social systems already about a decade ago.

Nowadays, smart phones are equipped with a wide range of sensors, including motion, location and environment sensors, that allow collecting rich observational data about human mobility in urban areas. Various predictive modeling tasks can be formulated based on such data. For example, one can be interested in recognizing the current activity of a person [11], predict next location [6], or predicting a trajectory of movement [13]. In this study, we explore challenges of preprocessing such sensory data for machine learning purposes for analyzing, modeling and predicting human mobility in urban areas. We present an experimental case study, report lessons learned and discuss challenges for future research.

The task of data preprocessing in mobile sensing is not trivial, and there are various challenges associated with this task. Data from sensors is collected as a sequence of time stamped observation records. Data records are not equally time spaced. Moreover, the timestamps of records from different sensors are not matching. In addition, observation records can be of different types: recording discrete events (e.g. battery charger plugged in), continuous processes (e.g. acceleration), or static measurements (e.g. current temperature).

The standard machine learning approaches for predictive modeling require data to be represented as instances. Instance (or example, case, or record) is defined as a single object of the world from which a model will be learned, or on which a model will be used (e.g., for prediction) [10]. However, data recorded by mobile sensors does not come as instances. Data comes as time stamped records, where time stamps are different for each sensor and are not equally spaced in time. The main data preprocessing question is, how to aggregate such data and convert it into instances for machine learning.

The problem of sensory data preprocessing is also not new, typically in the literature an arbitrary data aggregation approach is chosen and briefly mentioned (or not reported at all). However, there is a lack of dedicated studies focusing on the problem of preprocessing itself. Furthermore, the existing literature on preprocessing of mobile sensing data mainly deals with feature extraction from one sensor (e.g.

Table 1: Data collection rates (in sec.)

Source	Until June 14		After June 14	
	Per.	Dur.	Per.	Dur.
AccelerometerSensorProbe	-	-	30	30
ActivityProb	-	-	30	30
AndroidInfoProbe	86 400	-	86 400	-
BatteryProbe	1 800	-	1 800	-
BluetoothProbe	-	-	300	-
CellProbe	600	-	300	30
GravitySensorProbe	-	30	30	30
GyroscopeSensorProbe	120	-	30	-
HardwareInfoProbe	86 400	-	86 400	-
LightSensorProbe	1 800	30	120	30
LocationProbe	1 800	30	120	30
MagneticFieldSensorProbe	1 800	30	120	30
OrientationSensorProbe	120	30	30	30
RotationVectorSensorProbe	-	-	30	30
RunningApplicationsProbe	60	-	-	-
TemperatureSensorProbe	1 800	-	30	-
WifiProbe	600	-	300	-
DataUploadPeriod	3 600	-	3 600	-
Data annotation	manual		manual	

accelerometer or GPS signal) [5, 14], which is only one side of the problem. We are not aware of research works dealing with the problem of synchronizing data from multiple sensors.

This pilot study reports an exploratory case study in aggregating mobile sensing data and can be seen as the first step towards systematic treatment of this problem. We investigate how to construct instances out of sensory data for analyzing human mobility in three different scenarios: aggregating data with manual event annotations, converting static records to estimates of dynamic characteristics, and aggregating data from multiple sensors for predictive modeling purposes. The main contribution of this study is identification (via case study), and discussion of data aggregation challenges, as well as highlighting important questions for future research. To demonstrate the nature of mobile sensing data, we have released a data set called *Sensing Venice*, which is available as an open data set at the authors’ web-sites.

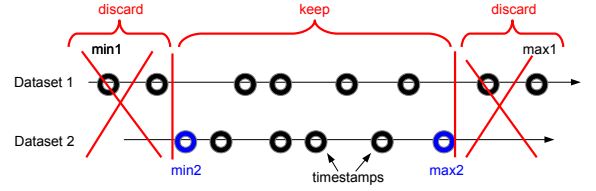
The paper is organized as follows. Section 2 presents a case study consisting of three experiments in analyzing mobility data. Section 3 summarizes the main challenges and lessons learned from the case study with respect to data preparations, and presents a taxonomy of settings and guidelines for data aggregation. Section 4 discusses open research directions and concludes the study.

2. EXPERIMENTAL CASE STUDY

We start the case study with a description of the settings and data, and then report methodologies and results from the three experiments. The experiments were selected to capture different nature of data aggregation challenges in processing mobile sensing records. The focus of this study is on data preparation techniques, therefore, we do not go further into domain driven analysis of the outcomes of the case studies and their implications from traffic management point of view. Domain-dependent treatise is left for future work.

2.1 Data collection and representation

Dataset has been collected using contextLogger3 [12], which

**Figure 1: Alignment of the time ranges.**

is an open source software tool for smartphone data collection and annotation based on Funf open-sensing framework [3]. Data was collected during the period from 2013 February 7 to 2014 January 27 using Sony Ericsson Xperia Active phone, which is using Android OS, v2.3 (Gingerbread). Summarizing our data collection upto January 2014, we have over 300 million timestamped records, resulting in approximately 13 Gb of data. The capacity of the battery is 1 200 mAh.

On June 14 the settings for data collection rates were changed. Table 1 provides details on sampling period and duration. Period indicates how often a given sensor is activated, and duration indicates for how long the sensor is activated. For example, if the period is 120 and the duration is 30 it means that the accelerometer is activated every 120 sec, and is collecting data for 30 sec.

We have released a mobile sensing dataset — coined *Sensing Venice* — as open data collection, which is available at <http://users.ics.aalto.fi/jhollmen/Data/>. This particular subset of data has been recorded in July 2013 in Venice, Italy. We hope that the mobile sensing dataset demonstrates the nature of the original raw data, and encourages other researchers to develop solutions to the challenges introduced in the current paper.

The experiments reported in this paper are based on the full six months dataset.

2.2 Experiment 1: processing event annotations

The first experiment investigates processing of event annotations. The start and the end time of an event is input by a user. These event annotations need to be cleaned, pre-processed and aligned with the recorded sensor data.

We illustrate these data preprocessing challenges by an experiment in modeling accelerometer data collection rate for different user activities. Accelerometer data is available only from June 14, hence, we use only that period of data in this experiment.

2.2.1 Methodology

We have two sets of recordings: event annotations and accelerometer records. Both are timestamped, but the timestamps are not aligned in any way. First we find the minimum (earliest) and the maximum (latest) time stamps in both sets, and discard the records from non-overlapping parts, as illustrated in Figure 1.

The main challenge in data preparation in this experiment is to extract activity labels from the event annotations. Annotations provide the start and the end time stamps of activities. Starts and ends are not necessarily paired, i.e., it may happen that there is a start, but no end, or there are three starts in a row and then one end of the same activity.

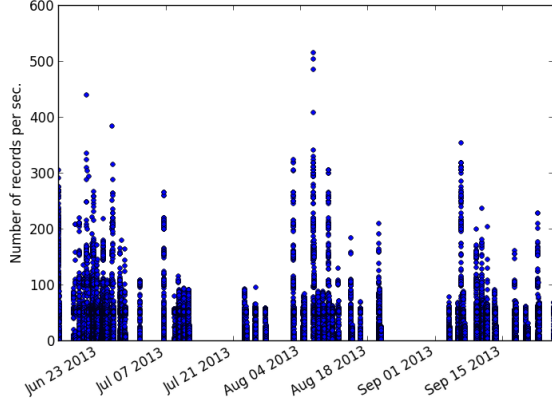


Figure 2: Accelerometer records over time.

We process annotations in a sequence. If there is a start, we consider an activity happening (no matter how many other starts of the same activity follow) until either of the following three triggers appear: annotation "stop", annotation "invalidate", or more than 6 hours have passed since the start. The latter rule is arbitrary chosen, assuming that mobility activities are typically short time.

For every second in time we create a label vector, where currently ongoing activities are encoded as 1, and not ongoing activities are encoded as 0. We get a label matrix \mathbf{A} of size $T \times k$, where T is the number of seconds from the beginning of data recording to the end, and k is the number of distinct activities recorded. Obviously, longer recording periods produce very large data files, therefore, one may consider choosing a larger time step for aggregation (e.g. creating a vector for every 10 sec. instead).

For modeling data collection rates, we need to process automatically collected accelerometer data and align it with the extracted activity labels. The time step, over which data is aggregated, needs to match the step used for label extraction earlier. We count the number of accelerometer records per second for every second that accelerometer was on. We get a vector \mathbf{X} of size $T \times 1$, where each entry is a number of records per second. Figure 2 shows the amount of data recorded over time.

Given the extracted label matrix \mathbf{A} and the record vector \mathbf{X} , we can obtain estimates for average records per second for each activity. There is an important modeling decision to be made. If two or more activities take place at the same time, how does it affect the number of records? Suppose activity a_1 generates n_1 records per second, and activity a_2 generates n_2 records. We could assume that if a_1 and a_2 take place at the same time, $n_1 + n_2$ records are generated. Alternatively, we can assume that if a_1 and a_2 take place at the same time, $\max(n_1, n_2)$ records are generated. In our experimental study we take the latter approach.

Following the first assumption, data collection rate can be modeled as a linear regression, where the inputs are binary indicators of activities, and the output is the number of records generated. If the second assumption is adopted,

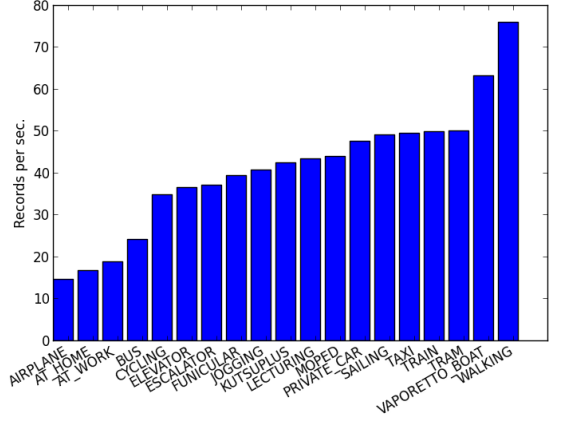


Figure 3: Average number of accelerometer records per second for each activity.

each activity is modeled independently, as follows:

$$\bar{r}_i = \frac{\sum_{j=1}^T a_{ji} x_j}{\sum_{j=1}^T a_{ji}}, \quad (1)$$

where i denotes the i^{th} activity, a_{ji} is the j^{th} entry of activity i in matrix \mathbf{A} , x_j is the j^{th} entry of vector \mathbf{X} .

Note, that this approach will automatically exclude the periods when the phone was off and no data was collected, since in those cases $x_j = 0$.

With this experimental setup we anticipate that different activities generate different number of accelerometer records. Raw sensor data in Android is acquired by monitoring sensor events. A sensor event occurs every time a sensor detects a change in the parameters it is measuring [1]. We expect different activities to have different acceleration patterns, and in turn to result in different data collection rates.

2.2.2 Results and observations

Figure 3 shows the resulting estimates of data collection rates for each activity. Data aggregated in such a way can be used, for instance, as a feature for activity recognition. While this feature stand alone would not be enough to separate all the activities, certain activities could be well distinguished, for instance, walking.

We see that walking produces the most records per time period, while at home or in the office activities produce the least. These results intuitively make sense. At home or office the phone would typically stay still on the table, hence, there is not much motion involved.

Moreover, we can see that conceptually similar activities appear close together, presenting similar amount of records. For example, "elevator" is very close to "escalator" and "funicular", where we would expect a smooth not too fast movement following a straight path. On the other spectrum of the scale "train" and "tram" appear nearby, both are means of transportation over rail. From this pilot experiment we can conclude that this preprocessing approach works and proceed to the next experiment.

2.3 Experiment 2: estimating the rate of change from static measurements

Sensors record static measurements; however, sometimes our interest may be to estimate dynamic characteristics. Examples include estimating speed of a moving object from GPS coordinates, estimating energy consumption from battery level indications, estimating flow rates from observed level of liquid.

The task in this experiment is to estimate how much energy is being consumed during data collection, given unequally time spaced observations of the battery level. The main challenges are: deriving conversion equations, filtering out uninformative observations, identifying and handling the periods with missing information (when the data collection application is off).

2.3.1 Methodology

For energy rate estimation we use level, voltage and status information from the BatteryProbe. Level indicates the percentage of battery charge remaining. Voltage indicates current voltage. Status indicates whether the phone is charging, discharging or the battery is full. All the records have the same time stamps.

Energy consumption in watt-hours (Wh) is computed as

$$E_{(Wh)} = Q_{(mAh)} \times V_{(V)} / 1000, \quad (2)$$

where Q is the electric charge in milliampere-hours (mAh), V is voltage in volts (V).

Given data recorded by ContextLogger2, the electric charge during the i^{th} time period, which starts at time t_i and ends at time t_{i+1} can be estimated as

$$Q_i = Q_{battery} \times (L_i - L_{i+1}), \quad (3)$$

where L_i and L_{i+1} are battery levels (in percentage) at the start and the end of the period.

However, there are two challenges. Firstly, data records are not equally spaced in time. As a result, time period i is not necessarily equal to $i + 1$ and, hence, Q_i is not comparable to Q_{i+1} . Secondly, battery levels are presented in low granularity (in rounded percents). As a result, estimation becomes stepped, where for several records the estimated energy consumption is zero (because $L_i = L_{i+1}$), then suddenly jumps and becomes zero again.

The first challenge can be overcome by estimating the rate of energy consumption instead of the amount of energy consumed. The rate of consumption is known as power P (in Watts), which during time period i can be computed as

$$P_i = Q_i \times 3600 / (t_{i+1} - t_i). \quad (4)$$

It is assumed that t is measured in seconds.

The second challenge can be addressed by discarding all the records of battery level, where the level remains the same as in the preceding record. This way we get less time intervals to consider, while the intervals themselves are longer.

2.3.2 Results and observations

Figure 4 plots the resulting energy consumption rate over time. We can see that most of the time energy consumption with ContextLogger is around 5 W. Negative energy appears when the phone is plugged for charging.

There are higher peaks of energy power, which may be due to switching ContextLogger on and off, when it is partially charged. In order to estimate energy more exactly at these



Figure 4: Estimated energy power estimates over time.

points, we would need to know or detect when context logger is switched on and off. Currently this information is not available from the logs.

Overall, from this pilot experiment we can conclude that it is possible to estimate the distribution of dynamic characteristics, such as energy consumption, from static sensor observations. However, this kind of preprocessing requires some domain knowledge input (e.g. knowing from physics how energy is defined). Nevertheless, we anticipate that it is possible to define a generic model form of such estimates for any sensor. This remains a subject of future investigation.

2.4 Experiment 3: data aggregation for predictive modeling

The goal of this experiment is to model energy consumption as a function of charging status of the battery.

2.4.1 Methodology

We model energy consumption as a linear function of indicator variables of the charging status: "discharging", "charging" and "full". We assume that energy consumption or inflow should be fully covered by these three sources; hence, we do not model the intercept (assume that the intercept is zero). With preliminary experiments using cross-validation we chose the Ridge regression optimization approach [7] ($\lambda = 1$) for finding the regression coefficients.

In the first experiment we discarded the observations, which did not indicate any change in the battery level. In this experiment we use all the records. We select an aggregation step s (e.g. 1 hour), which will be used to form data instances from raw observation records.

Energy consumption data is produced as specified in Algorithm 1. Voltage is estimated as $(V_i + V_{i+1})/2$. Energy power is estimated as in Eq. (4). We first divide all the time span into time periods of length s . Within each period we find all the observed records. We calculate energy consumption from record to record over time. Finally, we normalize the energy consumed from the actually observed time period to a fixed size time period s .

For example, if our period of aggregation is one day (24h), we may not necessarily observe records from 00:01 to 23:59.

It may happen, that we observe records only from 8:00 to 18:00. In such case, the factual time period is 10 hours. Hence, we would divide the observed energy consumption by 10 and multiply by 24 (the actual period of interest).

Algorithm 1: Aggregation of energy consumption data.

Data: A time ordered sequence of battery level L , voltage V , and timestamps t (N records); battery capacity $Q_{battery}$; aggregation step s (in sec)

Result: Dataset: energy consumption $\mathbf{E} = (E_1, E_2, \dots)$ (Wh) during time period s

```

1 for  $b = 1$  to  $(t_N - t_1)/s$  // number of bins
2 do
3    $E_b \leftarrow 0, T_b \leftarrow 0;$ 
4   for  $t_{now} \in [t_1 + (b-1)s, t_1 + bs - 1]$  do
5     // all time stamps within an interval
6      $E_b \leftarrow E_b + Q_{battery}(L_{now} - L_{now+1}) \times$ 
7      $\times (V_{now} + V_{now+1})/2000;$ 
8      $T_b \leftarrow T_b + t_{now+1} - t_{now};$ 
9   end
10   $E_b \leftarrow sE_b/T_b;$ 
11 end

```

Charging status data is aggregated in a similar way, as energy. For each time period b we have a three-dimensional vector of battery status, where each dimension indicates the percentage of time spent "discharging", "charging" or operating with "full" battery. The final dataset is a matrix with four columns, where the first three columns are the indicators of battery status, and the last column is the energy consumption. Each row corresponds to an observation period of 1 hour.

Since different sensors and sampling rates were active in the first and in the second period of data collection (before June 14 and after), we run the experiment in two parts, corresponding to these periods. For each period data is split into training and testing at random (50:50%). The regression parameters were estimated on the training part, and the model was tested on the testing part.

2.4.2 Results and observations

Table 2 presents the predictive models and their respective accuracies. The coefficients at the charging status mean the estimated energy consumption per hour. For example, 0.41 discharging means that when data is being collected and the phone is not plugged in, it consumes 0.41Wh of energy per hour. The negative coefficients mean that this is the net amount of energy the phone *gets*, when it is plugged into the electricity source.

We see that the directions of energy consumption (positive or negative) are identified correctly in both cases. In the second period discharging when a charger is plugged is excessively high (0.76), identifying reasons for that requires

further investigation. The relative magnitudes of energy consumption in the first period are convincing: charging is faster than discharging (0.89 Wh vs 0.41 Wh), and discharging when the charger is plugged is slower than when no charger is plugged (0.23 vs. 0.41 Wh). Interestingly, the energy consumption estimate is lower after June 14. It could be because of more inactivity periods during the second span; however, a further investigation is needed to analyze this phenomenon. Moreover, battery level is estimated rounded numbers, therefore the resulting energy consumption estimate is stepped and approximate.

Overall, from this experiment we conclude that it is possible to uncover, model and interpret relationships between processes with basic data aggregation; however, more investigation into accompanying data denoising is required, which remains a subject of future investigation.

3. DISCUSSION

The three case studies illustrate different challenges with data preprocessing. The first challenge is aggregating unevenly spaced and not synchronized in time observations, observed in Experiment 1. Given two sequences of observations, first we discard non overlapping (in time) parts, and then aggregate data over a fixed time step (1 sec).

Setting an appropriate aggregation time step presents one challenge for future investigation. The smaller the step, the faster the reaction time. However, the accuracy of the analysis may suffer if the step is too small to present an informative summary of what is happening. On the other hand, an excessively large time step only slows down the reaction time (e.g. a person starts walking, but recognition is delayed). Moreover, a large time step may capture heterogeneous data, for example, a mixture of several activities.

Another important open challenge is how to distinguish the periods of inactivity from the periods when no data is being collected, observed in Experiments 1, 2, 3. In this study we assumed that if there are no accelerometer records, then there is no activity. This is a crude approximation. Accelerometer sensor may be off or accelerometer sampling rate may be set to very large value (e.g. sample every 10 min). Failing to distinguish periods of inactivity from the periods when no data is being collected introduces noise in the resulting computational models. Such noise could be ignored, if there were only a few periods of inactivity or no data collection. However, when analyzing human mobility typically there are many more inactive periods than active periods. Unless a person is, for instance, a taxi driver, during a typical working day there would be several spans of movement and quite a lot of inactive periods, when the phone is resting in a bag or on a table. Therefore, reliable methods for filtering out the periods of no data collection and disambiguating the periods of inactivity need to be developed.

The next open challenge is how to deal with different granularity of sensor records, as observed in Experiment 3. For example, battery level is estimated in percentage, there are no decimals of percentage. If we are estimating energy consumption, the battery level would remain constant for a while before changing. If does not mean; however, that during that period no battery has been consumed. In Experiment 2 we overcame this challenge by introducing a variable data aggregation step, which varies depending on observed changes in the battery level. However, to deal with this challenge systematically we need some kind of smoothing

Table 2: Energy consumption as a function of battery status.

Period	Discharg.	Charg.	Full	MAE	R^2
Until June 14	0.41	-0.89	0.23	0.15	54%
After June 14	0.27	-1.66	0.76	0.19	72%

mechanism, that would also work online.

Finally, automatically processing manual annotations presents a big open challenge. Ideally, manual annotation of an activity should have a start and an end. In practice, an activity may have, for instance, multiple starts and no end, or an end, but no start. In addition, some activity time stamps may have manual corrections. In such a case end may happen earlier than the start, as we observed in Experiment 1. One way to deal with this challenge could be just to discard such corrupted data. However, manually annotated data is typically very scarce, therefore it is in the best interest of analysis to preserve as much of it as possible. Therefore, tailored data cleaning and imputation methods are needed. In our experimental investigation we introduced several simple intuition based rules to check and correct the integrity of user annotations. For example, if an activity starts and "end" annotation does not arrive for 6 hours, we consider the activity finished. A systematic generic approach to this problem is needed, that is a subject of future investigation.

4. SUMMARY AND CONCLUSIONS

We investigated how to aggregate mobile sensing data for machine learning purposes. We performed three exploratory experiments to illustrate different data preprocessing challenges. Following the experimental study, we identified and discussed several major challenges in mobile sensing data preprocessing for urban mobility analysis. The main directions are: how to determine the aggregation step, how to identify and isolate the periods of inactivity, how to deal with different granularity of observations, how to effectively automatically process manual data annotations, and integrate them with the observational data. To accompany the paper, we have released a subset of the data as openly available data, coined *Sensing Venice*. The data with its documentation is available at the authors' websites.

This pilot study sets a basis for further investigation aiming at producing a systematic methodology for preprocessing mobile sensing records for predictive modeling.

5. ACKNOWLEDGMENTS

This work was supported by the Aalto University AEF research programme and Academy of Finland grant 118653 (ALGODAN).

6. REFERENCES

- [1] Android developer's guide.
http://developer.android.com/guide/topics/sensors/sensors_overview.html.
- [2] Mobile economy europe 2013. Report, GSMA, 2013.
- [3] AHARONY, N., PAN, W., IP, C., KHAYAL, I., AND PENTLAND, A. Social fMRI: Investigating and shaping
- social mechanisms in the real world. *Pervasive and Mobile Computing* 7, 6 (2011), 643–659.
- [4] EAGLE, N., AND PENTLAND, A. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing* 10 (2006), 255–268.
- [5] FIGO, D., DINIZ, P. C., FERREIRA, D. R., AND CARDOSO, J. A. M. Preprocessing techniques for context recognition from accelerometer data. *Personal Ubiquitous Comput.* 14, 7 (2010), 645–662.
- [6] GAO, H., TANG, J., AND LIU, H. Mobile location prediction in spatio-temporal context. In *The Proceedings of Mobile Data Challenge by Nokia Workshop at the 10th Int. Conf. on Pervasive Computing* (2012).
- [7] HASTIE, T., TIBSHIRANI, R., AND FRIEDMAN, J. *The Elements of Statistical Learning*. Springer New York Inc., 2001.
- [8] HIMBERG, J., KORPIAHO, K., MANNILA, H., TIKANMÄKI, J., AND TOIVONEN, H. Time series segmentation for context recognition in mobile devices. In *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001)* (2001), pp. 203–210.
- [9] KHAN, W. Z., XIANG, Y., AALSALEM, M. Y., AND ARSHAD, Q.-A. Mobile phone sensing systems: A survey. *IEEE Communications Surveys and Tutorials* 15, 1 (2013), 402–427.
- [10] KOHAVI, R., AND PROVOST, F. Glossary of terms. editorial for the special issue on applications of machine learning and the knowledge discovery process. *Machine Learning* 30, 2/3 (1998).
- [11] KWAPISZ, J. R., WEISS, G. M., AND MOORE, S. A. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.* 12, 2 (2011), 74–82.
- [12] MANNONEN, P., KARHU, K., AND HEISKALA, M. An approach for understanding personal mobile ecosystem in everyday context. In *Effective, Agile and Trusted eServices Co-Creation — Proceedings of the 15th International Conference on Electronic Commerce ICEC 2013* (2013), vol. 19 of *TUCS Lecture Notes*, pp. 135–146.
- [13] MONREALE, A., PINELLI, F., TRASARTI, R., AND GIANNOTTI, F. Wherenext: A location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2009), KDD, pp. 637–646.
- [14] ZHANG, J., XU, J., AND LIAO, S. S. Aggregating and sampling methods for processing gps data streams for traffic state estimation. *IEEE Transactions on Intelligent Transportation Systems* 14, 4 (2013), 1629–1641.