
Anomalous Event Detection on Large-Scale GPS Data from Mobile Phones Using Hidden Markov Model and Cloud Platform

Apichon Witayangkurn

Institute of Industrial Science
The University of Tokyo
Komaba, Tokyo 153-8505, JAPAN
apichon@iis.u-tokyo.ac.jp

Teerayut Horanont

Institute of Industrial Science
The University of Tokyo
Komaba, Tokyo 153-8505, JAPAN
teerayut@iis.u-tokyo.ac.jp

Yoshihide Sekimoto

Institute of Industrial Science
The University of Tokyo
Komaba, Tokyo 153-8505, JAPAN
sekimoto@iis.u-tokyo.ac.jp

Ryosuke Shibasaki

Center for Spatial Information
Science
The University of Tokyo
Kashiwa-shi, Chiba 277-8568,
JAPAN
shiba@csis.u-tokyo.ac.jp

Abstract

Anomaly detection is an important issue in various research fields. An uncommon trajectory or gathering of people in a specific area might correspond to a special event such as a festival, traffic accident or natural disaster. In this paper, we aim to develop a system for detecting such anomalous events in grid-based areas. A framework based on a hidden Markov model is proposed to construct a pattern of spatio-temporal movement of people in each grid during each time period. The numbers of GPS points and unique users in each grid were used as features and evaluated. We also introduced the use of local score to improve the accuracy of the event detection. In addition, we utilized Hadoop, a cloud-computing platform, to accelerate the processing speed and allow the handling of large-scale data. We evaluated the system using a dataset of GPS trajectories of 1.5 million individual mobile phone users accumulated over a one-year period, which constitutes approximately 9.2 billion records.

Author Keywords

Anomaly Detection; GPS trajectories; Mobile Phone; Hidden Markov; Hadoop

ACM Classification Keywords

H.2.8 [Database Management]: Database Applications – data mining. H.3.4 [Information Storage and Retrieval]: System and Software – distributed system, clustering.

General Terms

Algorithms, Design, Experimentation, Performance

Introduction

With an increase in the urbanization, population growth, and changes in population density of many cities, understanding urban mobility patterns such as daily human activities and spatial temporal movements of populations are important aspects to explicitly express a current situation and allow an improvement in urban infrastructure. The increasing popularity of mobile phones embedded with positioning functionality such as GPS is allowing users to easily acquire their own locations and collect their own trajectories, which can then be used for various purposes such as location-based service applications. This has also led to the generation of massive spatio-temporal trajectory datasets. By analyzing such a large number of trajectories, the movement patterns of individuals and groups of people can be understood. Collective behavior is a term expressing the behavior of a large number of people, such as the actions of people gathering at a location for a social event. Mining such collective behaviors during a specific event allows a better understanding of how people act and respond during such times, which can then be used in

emergency responses and urban planning. For instance, when a typhoon hits a city, how people move, where they stay and in what numbers, and what are the most affected areas, are important pieces of information. Anomaly detection is the problem of discovering data patterns that are not similar to the expected behaviors and can be applied for detecting anomalous events by using collective behaviors such as the detection of temporal changes in population density within specific areas, which can either lead to or be an effect of a certain event. For example, a large movement of people may simply be the result of a large public fireworks display.

In our study, we focused on detecting anomalous events based on a spatio-temporal change in the population density. The study area was divided into equally sized square grids. We calculated the population density in each grid for each time period such as every hour. We then separated the data from each grid into seven groups based on the day of the week and applied K-mean clustering to cluster the population density into 10 clusters. Finally, we used hidden Markov model (HMM) to compute the patterns of each grid for each day. If the probability of the next sequence is much lower than the norm, it indicates that a change in the grid may have been triggered by a certain event. For this study, we used the trajectories of people in Japan that were accumulated over a one-year period and acquired from mobile phones using an embedded auto-GPS function sending out the user position approximately every five minutes. In this paper, we introduce the use of Hadoop, a cloud computing platform, to accelerate the processing speed, which can be used with real-world datasets rather than sampling data for research purposes. In

summary, the contributions of this paper are as follows:

- We propose a framework based on an HMM to detect anomalous area-based events and evaluate various parameters as features for improving the model, such as the number of points and unique users, and by adjusting the number of clusters and hidden states.
- We introduce the use of local scoring, or the difference in probability as compared with previous instances to detect a period when an event occurs.
- We purpose Hadoop/Hive, a cloud platform, with spatial processing functions, for processing large-scale datasets.

We evaluate the proposed method using a very large real-world mobile GPS dataset collected from approximately 1.5 million users in Japan over a year-long period.

Related Work

Mining the trajectories of people has become an attractive research field. Most works in this field have focused on extracting significant place of people [1][2], understanding human movement patterns [3], and predicting the movement of people [1]. Anomaly detection refers to the discovery of data patterns that are dissimilar to the expected behavior and to the detection of outliers. Anomaly detection is an important problem that has been studied in various research fields such as data mining and machine learning. It has also been used in many application domains such as intrusion detection, fraud detection and fault/damage detection [4]. Regarding anomalous trajectory detection, there are a number of previous works such as in Ref. [13] and [14]. However, in this work, we

focused on using changes in the population to detect anomalies generating from public events. Candia et al [5] reported that anomalous events influence human behavior and make people act differently from their usual patterns, but they did not state an exact method to detect such events. In addition, their dataset was considerably different from our research. In recent years, several works on anomaly detection based on crowd point distributions and point densities have been proposed. Pawling et al [6] reported the detection of anomalies using cell phones. They focused on data-clustering techniques to model the normality of the data; however, further research on the topic has stated that clustering techniques are quite meaningless in time-series sequences [7]. Liao et al [8] attempted to analyze the spatial distribution of moving points to facilitate the opportunity to detect abnormalities. PCA was used to remove the disturbed factors from a feature vector and maintain only the relevant information. Nevertheless, we used a grid-based system to calculate the features and apply the HMM to detect anomalies.

The research most related to our own is a work proposed by Yang et al. [9], who divided regions into small zones and counted the number of people in each zone, and then applied the HMM to model the probability of the sequences. However, our work has five major differences with [9]. First, Yang et al. used two datasets for their experiment: artificial data simulated using NetLogo software and real car-traffic data from loop detectors installed on freeways. However, our experiment focuses on real-world GPS data from mobile phone. The mobile phones used for our data collection have an embedded battery preservation function that deactivates the position

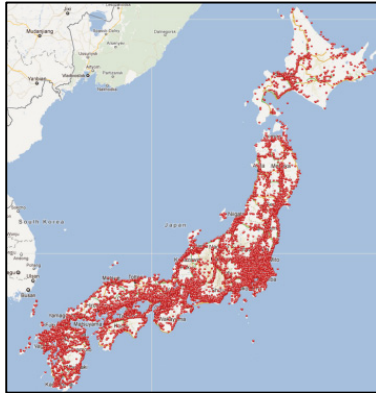


Figure 1. Data distribution in Japan

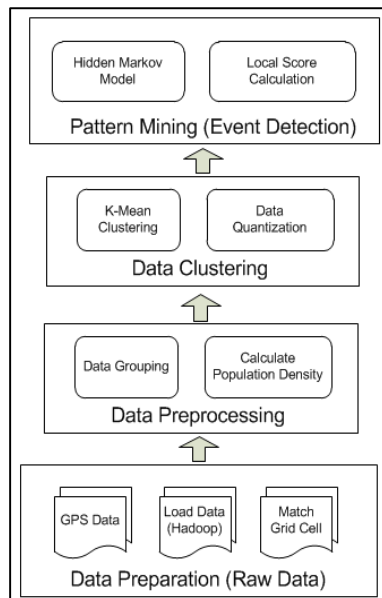


Figure 2. Overall framework of anomaly detection

sending function if no movement is detected. Hence, the amount of GPS data reflects the users' activities to a certain degree and our work is based on this assumption. Second, we used national grid data instead of defining our own zone, allowing us to map back to other useful properties surveyed by national departments, such as the estimated population in each grid, land use information, and trip information. As one example, if the system can detect a large train accident with in a specific area, the number of people affected can be estimated by combining the population data in that area with the number of people who typically used the railway for their transportation needs. Third, we applied a clustering algorithm to group the density data into smaller numbers, 1 through 10, making it easier to identify the level of population. This also simplifies the complexity of the HMM. Fourth, we aimed at finding an event and its time of occurrence, rather than only the day of the event, as presented in [9]. Finally, we propose the use of the Hadoop platform along with spatial techniques introduced in our previous work [10] to accelerate the overall performance of both data storage and processing speed.

A cloud computing platform is an excellent option for processing a large amount of data in the range of terabytes to petabytes with dynamically scalable and virtualized resources. Hadoop is an open-source large-scale distributed data processing that is mainly designed to work on commodity hardware [11], implying it does not require high-performance server-type hardware. Hive is a data warehouse running on top of Hadoop to serve in a data analysis and data query by providing a SQL-like language called HiveQL [12]. Hive allows users familiar with SQL language to easily understand and use query data. In a

performance comparison [10], Hadoop/Hive with enabled spatial capability produced very good results, reducing the processing time from 24 hours to 1 minute.

The GPS Dataset from Mobile Phone

The dataset was collected anonymously from about 1.5 million real mobile-phone users in Japan over a one-year period. A total of 9,201 million records were used. Data collection was conducted by a mobile operator and private company under an agreement with the mobile users. The positioning function included GPS activated on the users' mobile phones to send the current location data to the server every 5 minutes; however, several factors such as a loss of signal and the battery level affected the data acquisition. For example, the location-sending function was automatically turned off when no movement was detected. In addition, the geolocations were acquired and calculated from GPS, Wi-Fi, and cellular towers. Figure 1 shows the distribution of GPS data. To maintain user privacy, we used these datasets anonymously.

Overall Framework

Data Preparation

We employed Hadoop/Hive to store and process the data. In addition, we applied a spatial technique proposed in [10] to allow Hive to support spatial processing. In this step, we first loaded all GPS data stored in the CSV format into Hadoop through a Hive loading function. We divided an entire region into small grids of the same size. For standardization and compatibility, we used Japanese national grid in 500 m x 500 m grid shape. To associate a GPS point with a grid id, we developed a function in Hive to locate the

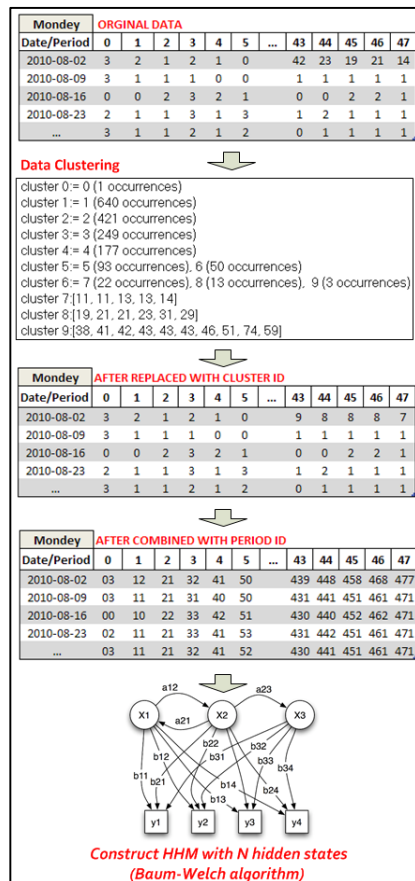


Figure 3. HMM for anomaly detection

point in a spatial polygon. Java Topology Suite (JTS), a spatial function library [16], was used to support the spatial function, and SR-tree spatial index was utilized to enhance the search speed [17].

Data Preprocessing

GPS data were attached with a grid id, as described in the previous step. For each grid id, the data were separated into seven groups based on the day of the week (Monday through Sunday) and time periods such as every 30 minutes or 1 hour. For each group and time period, two population density values were then calculated: the total numbers of points and unique users. As shown in Figure 3, the matrix rows show the dates, and the columns show the 30-minute time periods (a total of 48 periods were used). The outputs are stored in a multiple array column of a Hive table.

Data Clustering

To simplify the complexity of the model and make it more understandable, discrete observation values, rather than continuous observation values, were used. For each grid, K-mean clustering with $K = 10$ was applied to cluster the data in each group (matrix), resulting in 10 clusters. The cluster id was then labeled back to each value. For instance, the number of total points, ranging from 3 to 5, was assigned to cluster 3. All values with such ranges were replaced with cluster id = 3, as illustrated in Figure 3. We separated the clustering by each grid and group because we found that most people tend to have the same pattern on the same day of the week. For example, they go to work on each Monday using the same route. Hence, the distribution on the same day of the week is not too

diverse when comparing the clustering of all days together.

Pattern Mining for Event Detection

To handle pattern mining from grid-based data, an HMM was used to model each problem. The model parameters were trained using quantized observational data. The trained model was then able to calculate the probabilities of the new observation sequence and the possible state sequences. The trained model can also be used to predict unseen data or the next state. In addition, we used a local score, the difference in probability as compared with previous instances, to detect a period when an event occurs. For greater understanding, if the probability of the observation sequence is very low, it indicates that some events might have occurred on that day such as national holiday. For the local score, if the difference in probability of each period is very high, it means that such a case is not likely to have occurred and might have been caused by a special or anomalous event. Further details of this are described in the next section.

HMM for Anomalous Event Detection

In our approach, we constructed an HMM with 53 hidden states ($N=53$) and vector discrete observation values. The number of hidden states was selected based on our experiments. The observation values were a combination value of the period number and cluster id. The cluster id ranges from 0 to 9, which results from the data clustering step. For the observation sequences, we used $T = 48$ for every 30-minute period and $T = 24$ for every one-hour period. These two time periods were used to evaluate the possibility of detecting an anomalous event because our GPS dataset

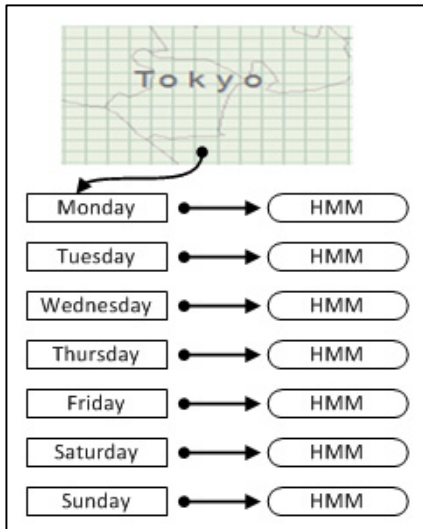


Figure 4. Grid-based level HMM for anomaly detection

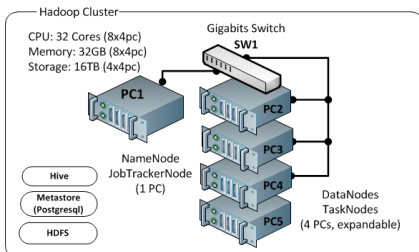


Figure 5. Hadoop cluster for data processing

is not so dense, i.e., one point for each five minutes, and sometime longer for less user activity. Zero-padding was also applied when no data were acquired during a particular period. For each time period, we built seven HMMs for each grid, as illustrated in Figures 3 and 4. The first HMM was for Monday and the seventh was for Sunday. A total of 961,257 grids were found in our dataset, or 63% of the entire country. In addition, approximately 6.7 million HMMs were constructed, and for accelerating the processing speed we decided to utilize a cloud platform. To find the best-fit model for each grid, we trained each HMM with its respective observation sequences using a Baum-Welch algorithm. The Baum-Welch algorithm tries to fit the model to most portions of the observation sequences regarded as normal activities or events. Rare anomalous events will therefore result in a very low probability of occurrence. Moreover, because the observation sequence range is quite long, it results in a very low probability value and might lead to an underflow problem. We therefore applied a scaling technique and used logarithmic values to avoid such a problem.

Local Score Calculation

In general, a new observation sequence is loaded into the HMM to find the probability that the input sequence will occur. For anomalous-event detection, if the probability of a sequence is very low on a given date compared with another, it indicates that a certain event may have occurred on that date. However, this method can only detect an anomaly for the entire sequence or at the date level, and not for the specific time period. To allow the time of an event occurrence to be detected, we used the concept of a local score. For an input sequence with a range of 48, we used the

forward-backward algorithm to calculate the natural logarithm of the probability of the given sequence at each period, and for each period, we then found the difference in the probability between periods t and $t-1$. If the difference is very large compared to that of the other periods, it indicates that a particular event may have occurred during that period.

$$\text{Local Score } (L_t) = \ln(\text{prob}(O_{ot})) - \ln(\text{prob}(O_{ot-1}))$$

where $O_{ot} = \{O_0, O_1, \dots, O_t\}$ is a subset of the input sequence from time 0 to time t .

Evaluation

Number of points vs. number of unique users

We calculated two types of observation values. One is the total number of points and the other is the total number of unique users in each respective grid and period. From our experiments, we found that the total number of points gave better results than the total number of unique users, with significant differences in certain grids, such as those where a train station is located. One reason for this is the movement-detecting function used for sending a GPS point. For our dataset, even though the data-sending interval was set to five minutes, the GPS data were sent only when movement was detected. This function was used to preserve battery usage. For example, there will be no data during the nighttime when users are sleeping at home. For event detection, when an event occurs, people may conduct more activities than usual, leading to an increase in the number of GPS points. For a more definite example, if a train has stopped at a station after an accident, passengers will be unable to travel to another location for a certain period of time. The

Testing Platform

Hadoop Cluster: Our Hadoop cluster, shown in Figure 5, consists of five computers with the same specifications, a 2.6 GHz 8-Core Xeon CPU, 8 GB of memory, and two 2TB hard drives. CentOS 6.0, 64-bit, was used as the operating system.

Implementation: We used Java for the development language. Java Topology Suite (JTS), which is a Java-based spatial library, was used for supporting spatial calculations such as finding geometry points and spatial indexing for fast geometry searches. For data mining techniques, we used the Java Machine Learning Library (Java-ML) for clustering, feature selection, and classification. We developed a function on Hive to calculate all necessary features as well as the probability values using a user-defined function (UDF).

number of unique users may not increase greatly because the transportation mode has been blocked. On the other hand, the number of points increases a great deal because people may move around the station or play with their mobile phones while waiting.

Processing Performance

Figure 6 illustrates the processing time of a single computer as compared with a Hadoop cluster. With four nodes, Hadoop showed a significant improvement over a PC, using only several hours to process all datasets. Additionally, this processing time did not include data preparation and data preprocessing steps. Based on a performance comparison in our previous work [10], it takes months for the process of data preparation in a PC.

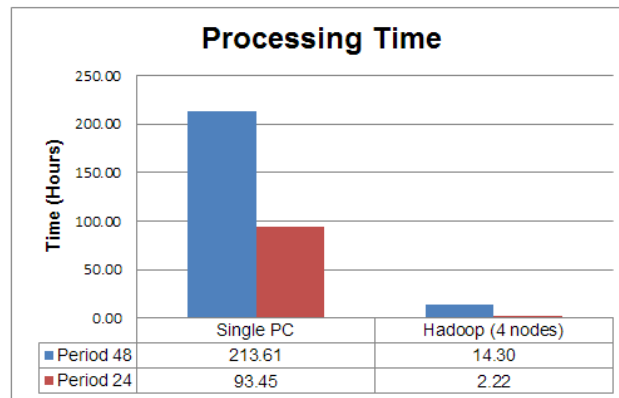


Figure 6. Processing time
Experimental Results

To demonstrate the results of our system, we selected several known events to interpret data during our experiment. The first event selected was the Edokawa fireworks festival held on August 7, 2010. We used a

trained HMM model to calculate the probability of the sequence. As shown in Figure 7, the probability of the festival day was much larger than for other days, which typically have a value of approximately 60. This result indicates that the total probability or full sequence probability can be used to detect anomalous events, particularly for long events at the date level because shorter events may not significantly affect the probability.

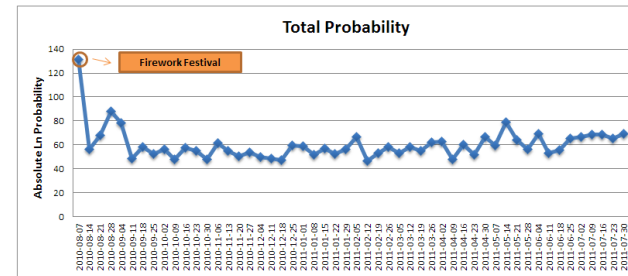


Figure 7. Probability of full sequence at Edokawa

Another example is a grid near Akihabara (an electronics shopping area). Using only the total probability, the proposed system can still detect national holidays and some certain other events such as the effects of the Great 3.11 Earthquake, as shown in Figure 8.

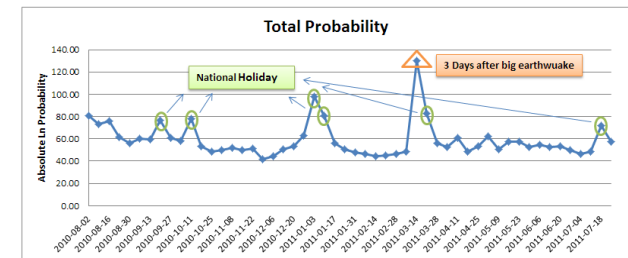


Figure 8. Probability of full sequence at Akihabara

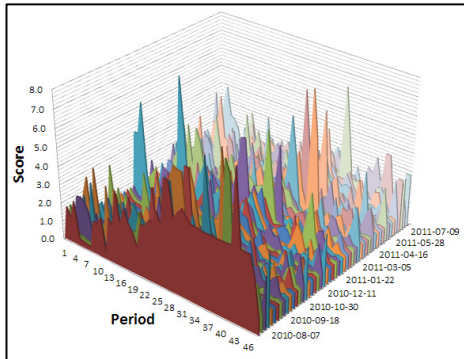


Figure 9. Local score comparison at Edokawa

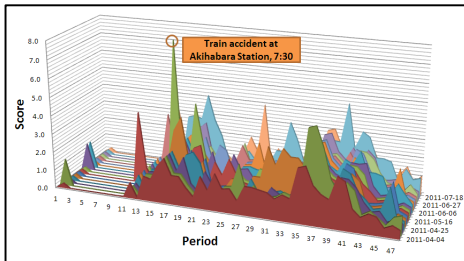


Figure 10. Comparison of local scores for a train accident event

Because the full-sequence probability can only detect anomalous events at the date level, we used local score instead to identify shorter event during the day. The graphs illustrated in Figure 9, clearly shows that the day of the fireworks differed greatly from other days. It also indicates anomalous periods; in this case, anomalies started from late morning and peaked during the afternoon. Figure 10 shows the results for a train accident event at Akihabara station. The local score peaks at 7:30 (15th period), which was the time of the accident. All other trains passing that area also had to stop for 30 minutes. We also evaluated local score technique with 56 events including firework events, New Year events and earthquake event. We found that with a local score value of more than 3.0, all event could be clearly detected as anomaly event. Furthermore, we plotted the results using a grid polygon on a map to see the overall view of a large event. We used a local score with a threshold of 3.0 to demonstrate the anomalous event detection for large areas. Hence, if the score is higher than 3.0, a red block will appear on the map. Figure 11 shows the results on the day of the Great 3.11 Earthquake in the greater Tokyo area. As shown in Figure 11(b), anomalous events were detected in many areas. In this case, it is possible to apply this technique to find the affected areas based on certain events. A wider view is illustrated in Figure 16. A number of anomalies were detected in many areas, most of which were affected by the earthquake, such as Sendai, Ibaraki, Fukushima, and Tokyo.

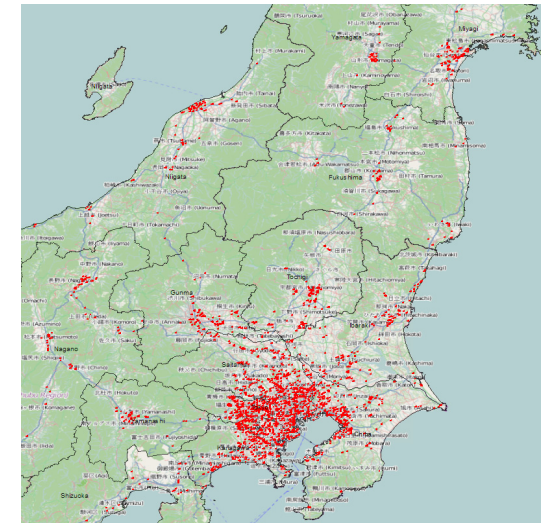


Figure 12. Wider view of the period just after the Great 3.11 Earthquake

Conclusion

In this paper, we proposed a detailed framework including a scalable development platform for detecting anomalous events from large-scale mobile GPS data. The framework consists of four steps: data preparation, data preprocessing, data clustering and pattern mining for event detection. K-means clustering was applied to quantize the observation data. An HMM was used as the main algorithm for pattern mining. Together with the HMM, we introduced a local score to detect the specific period of an anomalous event. For the observation feature, particularly for our dataset, we used the number of points as a feature because this gives better results compared to the number of unique users. The experimental results showed that the HMM

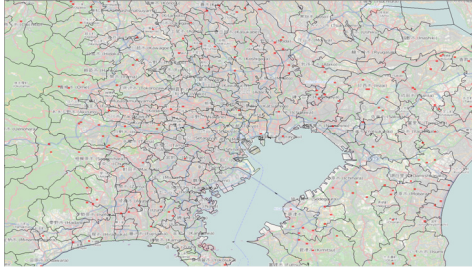


Figure 11 (a). A period before the Great 3.11 Earthquake

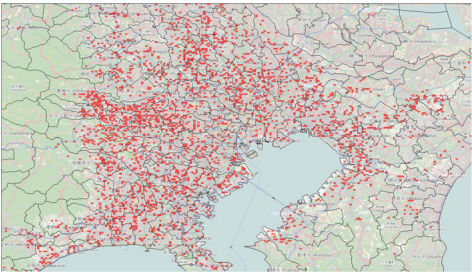


Figure 11 (b). A period just after the Great 3.11 Earthquake

did very well in pattern creation, as well as in detecting anomalous events. Using full-observation sequence probability, a lengthy anomalous-event period on the same day such as a national holiday can be detected. Through the local score, it is possible to detect an event down to the level of the event period, rather than only at the date level. The proposed system clearly distinguishes periods of anomalous events from other periods. Additionally, for large-scale data processing, we utilized Hadoop/Hive, a cloud computing platform used as a data-storage system, to speed up the processing time. The results show that Hadoop uses only approximately 6% of the time required for the computer to finish processing.

Acknowledgements

The work described in this research paper was conducted with an agreement from Zenrin Data Com to use mobile phone datasets of personal navigation service users. This work was supported by GRENE (Environmental Information) project of MEXT (Ministry of Education, Culture, Sports, Science and Technology).

References

- [1] Ashbrook, D., Starner, T. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing* (2003), 7(5), 275-286.
- [2] C. Zhou, et al. Discovering Personally Meaningful Places: An Interactive Clustering Approach. In *ACM Trans. on Information Systems* (2007), vol. 25(3).
- [3] Liao, L., et al. Building Personal Map from GPS Data. In *proceedings of IJCAI MOO05*, Springer Press (2005), 249-265.
- [4] Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection. *ACM Computing Surveys* 41, 3 (2009), 1-58.

- [5] Candia, J., Gonzalez, M.C., Wang, P., Schoenharl, T., Madey, G., and Barabasi, A.-L. Uncovering individual and collective human dynamics from mobile phone records. *Journal of Physics A: Mathematical and Theoretical* 41, 22 (2008), 224015.

- [6] Pawling, A., Yan, P., and Candia, J. Anomaly detection in streaming sensor data. *Intelligent Techniques for Warehousing and Mining Sensor Network Data*, (2008), 99-117.

- [7] Keogh, E., Lin, J., and Truppel, W. Clustering of time series subsequences is meaningless: implications for previous and future research. *Third IEEE International Conference on Data Mining*, (2003), 115-122.

- [8] Liao, Z., Yang, S., and Liang, J. Detection of Abnormal Crowd Distribution. *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, (2010), 600-604.

- [9] Yang, S. and Liu, W. Anomaly Detection on Collective Moving Patterns. *IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing*, (2011), 291-296.

- [10] Witayangkurn, A., Horanont, T., and Shibasaki, R. Performance comparisons of spatial data processing techniques for a large scale mobile phone dataset. *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications - COM.Geo '12*, (2012), 1.

- [11] Hadoop Project: <http://hadoop.apache.org/>

- [12] Hive Project: <http://hive.apache.org/>

- [13] Chen, C., Zhang, D., Castro, P.S., et al. iBOAT: Isolation-Based Online Anomalous Trajectory Detection. *IEEE Transactions on Intelligent Transportation Systems* 14, 2 (2013), 806-818.

- [14] Xiaolin, L., Chawla, S., Liu, W., and Zheng, Y. On Detection of Emerging Anomalous Traffic Patterns Using GPS Data. (2012).