iTaxi: Context-Aware Taxi Demand Hotspots Prediction Using Ontology and Data Mining Approaches

Han-wen Chang **Yu-chin Tai** **Hsiao-wei Chen** **Jane Yung-jen Hsu**[*]

Department of Computer Science and Information Engineering
National Taiwan University, Taiwan
{r96005, r96047, r96020, yjhsu}@csie.ntu.edu.tw

Abstract

It has been estimated that over 60 thousand licensed taxis in the Great Taipei area are not occupied over 70 percent of driving time on average. However, the taxi company, TaiwanTaxi, indicates that even in rush hour, there are customers whose requests are not satisfied. The demand and supply are not paired, causing not only customers wait too long for a cab, but also taxi drivers waste time and fuel to wander around the streets.

In this paper, it uses spatial statistics analysis, data mining and clustering algorithm on historical data of taxi requests to discover the demand distribution, which varies from different environment contextual information such as the location, time, and weather. Finally, the predicting system then predicts potential hotspots of taxi requests and provides hotspots information for drivers to reduce vacant time of the taxi.

Keywords: Data Mining, Clustering, Association Rule Mining

1. Introduction

According to the Institude of Traffic (IOT) Survey of Taxi Operation Conditions in Taiwan Area 2006 [1] , in average, each taxi driver operates the business 9.9 hours a day, driving approximately 147.3 kilometers. However, about one-third of time (3.2 hours), drivers are on the roads without passengers. The time and energy wasting phenomenon is more severe in Taipei area. Taipei City Department of Transportation [2] reports that in over sixty percent to seventy-three percent of the their operation hours, taxi drivers are driving without passengers. This roaming situation not only wastes energy but pollutes the environment. One of the reasons for driving an unoccupied vehicle is that they do not know where potential passengers are, leaving them with no choice but to wander around the city.

To solve the problem, understanding and constructing the model of the demand is important. Analyzing the data on past history, including the time and location passengers get on taxis, provides clues to the demand distribution. Given the time and environment context, relevant records are filtered for further computation. Clustering methods can be applied on primitive data to find locations with high density. Mapping these clusters to known road segments helps in our understanding of the semantic meanings of the geometries. Once the clusters are identified, the hotness index can be calculated. Combing the cluster geometries, the semantic road names, and the hotness index, hotspots are defined. As a consequence, drivers can adjust their strategies.

The remainder of this paper is organized as follows. Section 2 describes the related work. The problem formulation is presented in Section 3. Following the definition, Section 4 details our approach. Next, Section 5 briefly describes the implementation and experiment results. Finally, concluding remarks and future directions are stated in Section 6 and Section 7.

2. Related Work

Some researches focus on the analysis of popular places, which are so-called hotspots. CrimeStat [4] is a spatial statistics program which helps police to analyse crime incidents. It provides an interface for human to interact with computer, choosing different clustering methods like k-means [5], hierachical clustering [6], or kernel density estimate to cluster crime incidents. It helps user to visualize crime incidents so as to discover crime hotspots. In London, there is a statistical data for road accidents. Anderson [1] points out that there is no universal definition of a hotspot for road accidents, and comparing serveral methods for discovering hotspots. Clustering techniques [12] are widely used for grouping similar items. X-means [8] is an extented version for k-means, and it can decide the value of k itself by using Bayesian Information Criterion (BIC) [9].

There are researches focusing on finding significant locations from GPS traces. Ashbrook and Starner [2] use k-means-like iterative approach to cluster places into locations. Palma et al. [7] propose a clustering method based on speed measurement to distinguish stops and moves in a single trajectory. This work

[*] Correspondence author

[1] http://www.iot.gov.tw/public/Attachment/71081642971.xls

[2] http://www.dot.taipei.gov.tw/

is different from the above in that GPS trace records have strong spatio-temporal continuity, but taxi request records are not. In addition, GPS traces are from individuals and are more personalized, while taxi requests are with less personalized factor.

Spatial co-location pattern mining is to find the set of spatial features that are frequently located together in spatial proximity [10]. To find co-location rule with high prevelance and high confidence, approaches similar to association rule mining are used. This work is different from co-location pattern mining because spatial features are not the only domain considered. Context of temporal features are involved to find more specific context-dependent patterns. Modifications are required to formulate the context-aware pattern mining problem into association rule mining problem.

The contribution of this work is the application to solve the context-aware pattern mining from taxi request records by adapting existing approaches from clustering and association rule mining. Through the process, customer demand can be understood.

3. Problem Definition

For taxi drivers roaming on the road and looking for potential customer, finding the nearby candidate positions to wait is the first task. Based on a reference position, the weather condition, current time, the request history and the location model, hotspots around the reference position can be predicted and recommended. With the analysis result, taxi drivers can adjust their strategies and decide where to go to pickup passengers. The following representations are used to formulate the problem.

The primitive contexts in this work involves the location, time, the weather condition. Latitude and longitude, denoted as ϕ and λ respectively, are used as the coordinate system to specify the geometry of locations. The weather condition, denoted as w, says whether it is raining, and the instant time t in calendar clock. Time intervals and relationships among them are defined in the ontology, see Fig. 1.

The location dataset D_L stores m_p landmarks and m_r roads. Each landmark or road in the dataset is defined with its identical name $name_i$, the geometry $geom_i$ within the given coordinate system, and the category cat_i it belongs to. The geometries of landmarks are mostly defined as the representative points, while the roads are mainly as polylines. On these locations, geometry relationship functions, such as `COVERS`, and processing functions, such as `DISTANCE`, are defined. `COVERS` indicates whether one geometry fully covers the other geometry; `DISTANCE` calculates the shortest distance from one geometry to the other. With these properties, the hierarchical structure of these locations in the geospatial domain can be established.

The categories and the semantic relationships are defined with the location ontology. The categories are given on the basis of the functionalities of the landmarks or the classes of the roads. Functionalities are organized in a hierarchical structure; each functionality is represented as a string code. For example, tourist spots (500) are separated from goverment offices (100) and schools (200), and the spots can be further divided into subcategories such as recreational parks (505) and night markets (511). Part of location model in the representation of ontology is shown in Fig. 2.

$$D_L = \{p_1, p_2, \ldots, p_m\}, m = m_p + m_r$$
$$l_i = (name_i, geom_i, cat_i), i = 1 \ldots m$$

The request history dataset D_R stores n past taxi request records. A single taxi request record r_i contains the position including latitude ϕ_i and longitude λ_i, the timestamp $time_i$ when the customer gets on the taxi, and the weather condition w_i at that time.

$$D_R = \{r_1, r_2, \ldots r_n\}$$
$$r_i = (\phi_i, \lambda_i, t_i, w_i), i = 1 \ldots n$$

The passengers making the requests may previous leave the landmarks on the same road, but get on the taxis at different nearby positions. With the imprecision of GPS signals, these request records will not be identical but spatially close to each other. As a result, these nearby records are grouped into clusters, and these clusters can be further mapped to roads or landmarks which cover most of the points in the cluster. Hence, the semantic meaning of the cluster can be represented by the roads or landmarks.

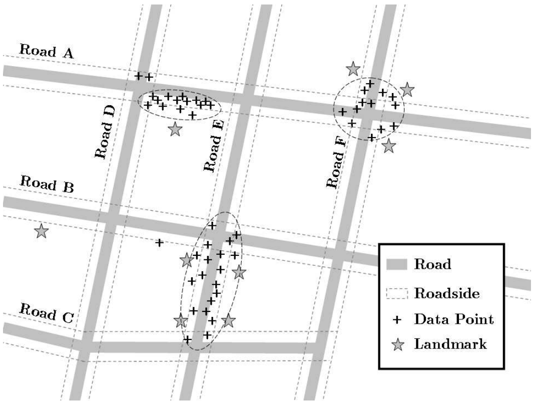


Fig. 3. Roads, landmarks, and request records

Take Fig. 3 for example. There are six roads (the bold lines), forming nine junctions, and nine landmarks (the star-shape points) in this illustration. Each road can be divided into several road segments in respect to the junctions. For each road segment and junction, the roadside is formed by adding a buffer distance to the corresponding geometry (the dash line areas). In this illustration, the request records (the plus-sign points)

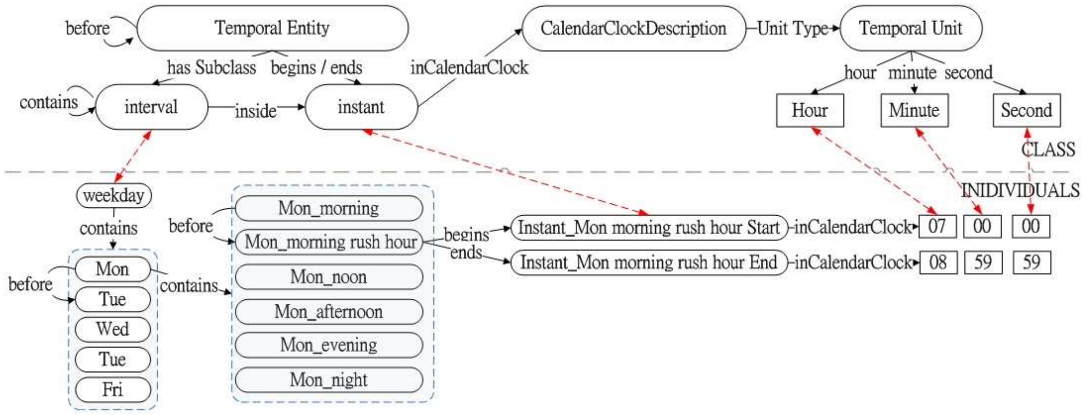


Fig. 1. Part of the time ontology

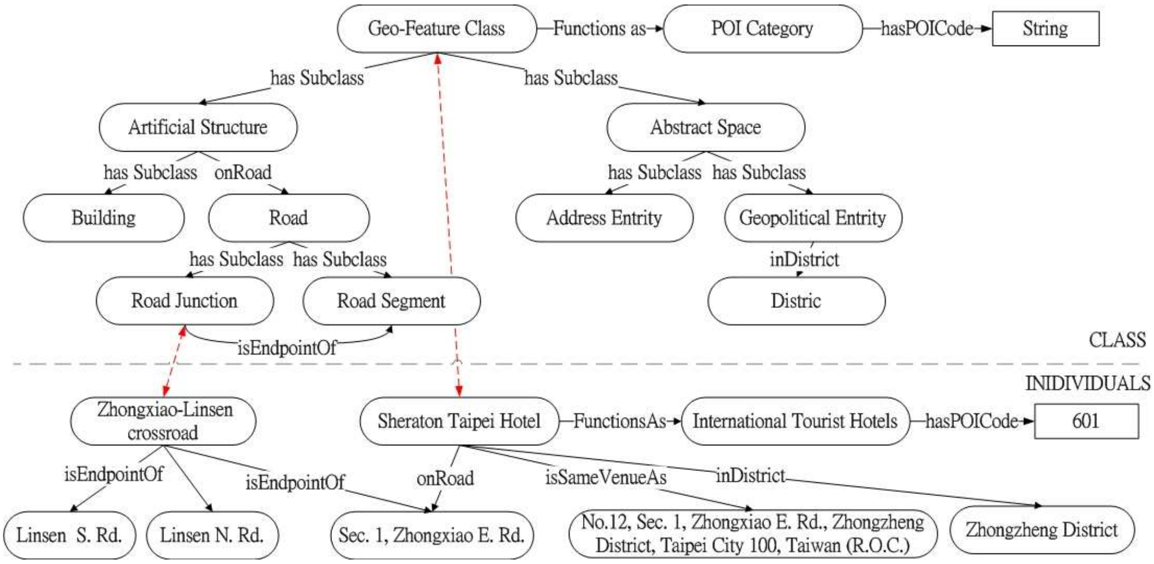


Fig. 2. Part of the location ontology

obviously form three main clusters (with few outliers). The upper-right cluster can be represented as “at the intersection of road A and F,” while the lower cluster can be represented as “on the road E between road B and C.”

When the system detects the need of prediction, such as when the latest pessenger gets off the taxi, the contexts are compiled as a query for the hotspot prediction. A query for the hotspot prediction $Query_t$ involves the reference position, weather condition and time. The position (ϕ_t, λ_t) is the latitude and longitude of the reference point of the taxi; the weather condition w_t says whether it is raining, and the query time is characterized by the day of week d_t and the hour of the day h_t.

$$
\begin{aligned}
&Query_t = (\phi_t, \lambda_t, w_t, d_t, h_t) \\
&\quad w_t \in \{Rainy, Clear\} \\
&\quad d_t \in \{Mon, Tue, Wed, Thu, Fri, Sat, Sun\} \\
&\quad h_t \in \{0, 1, 2, \ldots, 23\}
\end{aligned}
$$

The expected output of the query $Query_t$ with the request history and location dataset D_R, D_L is a set of hotspots H which are composed of the geometries of clusters, the semantic names as roads or landmarks, and the corresponding scores $s_i \in [0..1]$ indicating the degree of hotness.

$$
\begin{aligned}
H &= \{h_1, h_2, \ldots, h_k\} \\
h_i &= (C_i, name_i, s_i), i = 1 \ldots k
\end{aligned}
$$

4. Approach

Considering the case when a taxi driver is taking a passenger to the destination. When the taxi driver approaches the location and drops off the passenger, the system detects the need of the driver to know the potential taxi demand. As a consequence, the prediction service begins and the results will be displayed to the driver for reference. Fig. 4 shows the flow of the approach we used.

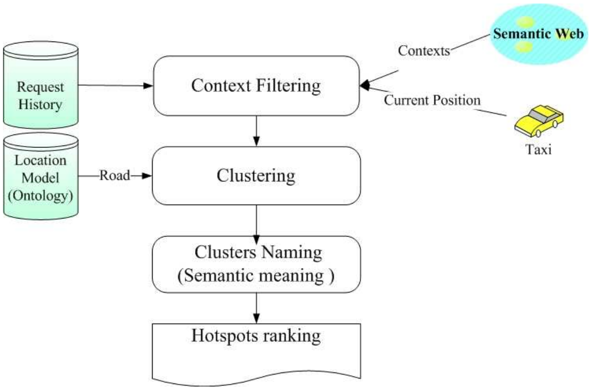


Fig. 4. System Architecture and Flow

According to the contexts, request records from request history dataset are retrieved and filtered; these records are later spatially grouped into clusters. For each cluster identified, the road which fits the distribution of the cluster is found and used to annotate the cluster, giving semantic meanings for understanding. Then, considering the number of requests during the time span, the areas of the clusters, and the distances from the position of the driver, the hotness scores of the clusters are calculated. The geometry of the cluster, the corresponding semantic meaning, and the hotness score together forms one hotspot.

4.1 Context-based Filtering

Context-based filtering, the first step of the whole flow, picks out the relevant request records for calculating hotspots. The records with exactly the same contexts will be selected first and form the dataset for clustering. If the amount of the dataset is not large enough, the context constraints will be relaxed to a super-concept according to the context ontology. For example, 7:30 AM, Monday can be relaxed to Monday morning rush hour (Fig. 1). And all records under the relaxed context will be considered for the following steps.

4.2 Clustering Positions

Clustering the GPS coordiates into locations is an important step before doing any further analysis. On spatial dimension, millimeter-level scale is too detailed to make comprehensive conclusions for real applications. In the scenario of analyzing taxi demand, city-block-level or road-level scale with semantic meaning is much easier to describe the distribution of request records. Passengers coming from a business building, which may be a hotspot for taxi, may actually get on the taxis at slightly different GPS coordinates on the roads around the building. These nearby GPS coordinates should be viewed as one location instead of several independent locations.

4.2.1 Similarity Measures

Suppose the two points a, b are at $a = (\phi_a, \lambda_a)$ and $b = (\phi_b, \lambda_b)$. The most common distance measure of two points on the map is the Euclidean distance. However, the earth is roughly a great circle, and the latitude and longitude are defined globally in respect to the earth surface instead of a plane, the Euclidean distance is not an accurate measure and the scaling parameter depends on the latitude value. In our work, Vincenty's formula [11] is used with the assumption of spherical Earth. The radius of the Earth is assumed to be 6372.795 kilometers, and the geodesic distance between two points is the radius times the angular distance $\Delta\widehat{\sigma}$ which is given in the Equation 1.

4.2.2 Clustering Algorithm

The selection of similarity measure and clustering algorithm decides the result of clustering. There are several clustering algorithms nowadays, and each has its pros and cons when facing different kind of data. No single algorithm outperforms for all the problems. In this work, three clustering algorithms were tried, and the details are described in the implementation section.

4.3 Mapping Clusters to Roads

Each cluster contains several nearby request records, and the next step is to give the clusters semantic meanings. Assigning a good semantic meaning to one cluster without any reference or attribute properties is almost impossible. In this work, the roads which match the clusters are identified, and the meanings of the clusters are assigned as the names of the roads.

Roads may intersect with each other and form several junctions. Breaking at the junctions, roads can be divided into contiguous road segments. The road junctions and road segments are the smallest unit in the location model, and the hierarhical structure is defined in the ontology.

For each cluster, the system lists the road segments inside the convex hull of the cluster. If there is only one road segment found, the name of the road segment is used to annotate the cluster. If there are more than one road segments, contiguous segments are connected to derive longer roads, and the longest road formed by the road segments is used.

4.4 Predicting Hotspots

For taxi drivers roaming on the road, good prediction of hotspots is time-saving. According to different contexts such as time, date, weather, and the position of

$$
\begin{aligned}
dist(a,b) &= 6372.795 \times \Delta\widehat{\sigma}(a,b)\\
\Delta\widehat{\sigma}(a,b) &= \arctan\left(\frac{\sqrt{(\cos(\phi_b)\sin(\Delta\lambda))^2 + (\cos(\phi_a)\sin(\phi_b) - \sin(\phi_a)\cos(\phi_b)\cos(\Delta\lambda))^2}}{\sin(\phi_a)\sin(\phi_b) + \cos(\phi_a)\cos(\phi_b)\cos(\Delta\lambda)}\right)
\end{aligned}
\tag{1}
$$

taxi driver, the prediction system should discover which area is the hottest. To calculate the hotness index of the hotspot h_i, four parameters are defined. The notation ρ_{dist} is the distance between the hotspot and the taxi location; hotspots closer to the taxi position are preferred. The number of points inside the cluster is ρ_{num}; the cluster area, ρ_{area}, is calculated according to the convex hull formed by the points of the cluster. The time span, ρ_{time}, of the filtered request records is considered to convert counts into rates. For example, if the records are from morning rush hour, 7 AM to 9 AM, during the two month period, the time span is 120 hours (two hours per day times sixty days). A normalized factor, η, is used to adjust hotness index. Therefore, the hotness index s_i of the hotspot h_i is defined as Equation 2.

$$
s_i = \eta \times \frac{1}{\rho_{dist}} \times \frac{1}{\rho_{area}} \times \frac{\rho_{num}}{\rho_{time}} \tag{2}
$$

4.5 Finding Associations Rules

After filtering the request records, clustering the hotspots, associating the semantic meanings, and calculating the hotness index, the context-dependent association rules can be established. One association rule is formed as $\{w_t, d_t, h_t\} \rightarrow h_i$ where the left hand side is the set of context, including the weather condition, the day of the week, the hour of the day, and the right hand side is the hotspot with its geometry, semantic meaning, and hotness index. The meaning of the rules itself is that under the given context, requests are probably happend at where the hotspot locates. With these context-dependent association rules, the hotspot phenomena can be explained and understood in a more descriptive way.

5. Implementation and Experiment

The location dataset and request history dataset are the fundamental components of the system; the quantity and quality of the datasets affect the performance of the system. In the following, the dataset collection process is described in details. On the request history dataset, three clustering algorithms were applied. The results are shown, and the comparison is provided.

5.1 Location Dataset

The current location dataset is built based on the research data version 1.4 provided by the Institude of Transportation (IOT), MOTC. In the research data, landmarks and roads are provided with their geometries and relevant attributes. In this work, only the data describing Taipei, Taoyuan, Hsinchu, Miaoli, Keelung, and Ilan are considered; a total of 11,750 landmarks and 179,772 road segments are imported.

5.2 Request Data Collection

The data collection process is supported by the Institude for Information Industry in collaboration with Taiwan Taxi Company. Five taxi drivers, mainly operating their business in Taipei city, were asked to record when and where passengers getting on their taxis from June 25 to August 25, 2008. The identities of these drivers are represented as numbers, such as #1000. Each taxi driver was given a PDA and a bluetooth GPS receiver. When passengers get on the taxi, the driver selects the mode and weather on the PDA screen and make records. Time and GPS coordinates information are directly copied from GPS receiver. These records are stored on the PDA during the collection period. During the two month period, 2,319 request records were collected. However, 487 records were ignored because their GPS readings are zeros or out of range.

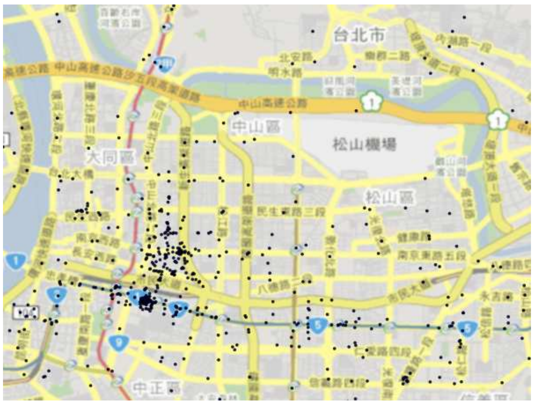


Fig. 5. Spatial distribution of real requests.

The spatial distribution of the real taxi requests are shown in Fig. 5; the figure on the right is the zoom-in version. In this figure, clusters of points around Shandao Temple and Linsen North Road are obvious. The

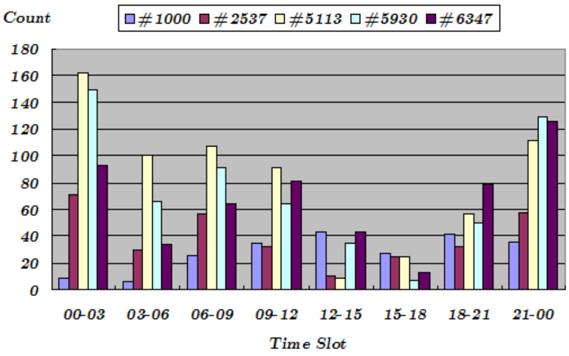


Fig. 6. Temporal distribution group by drivers.

temporal distribution of each driver may be different (See Fig. 6). Four of five drivers mainly operated in the late evening to the early morning, while the taxi driver #1000 took more passengers in the afternoon than midnight.

5.3 Clustering Implementation

Clustering analysis has been on focus for a long time. Several approaches are developed and improved year by year. In this work, three clustering algorithms are implemented to see which algorithm fits the request records distribution.

5.3.1 K-means

K-means [5] is the most common hard partition clustering. At first, it must be given a fixed number k to determine how many clusters should divide into. After that, it starts to do iterations to reassign each item into k clusters. Iterations will be stopped when the cluster members do not change or the change of each cluster mean is small enough. The advantange of k-means consists in its easy implement and local minimal convergency. However, it has some drawbacks. Firstly, the number k cannot be decided by itself. Secondly, the result of k-means may be changed if initial points are not the same. Thirdly, Outliers will affect damatically to the result of k-means.

5.3.2 Agglomerative Hierarchical Clustering

Agglomerative hierarchical clustering [6] groups similar clusters/objects together to form high-level clusters in each iteration. After grouping all individual objects into one final cluster, a binary-tree structure is created. Given a cut-off threshold of maxima distance, agglomerative hierarchical clustering returns the cluster sets that none of the distance between two clusters is smaller than the threshold. Hence, an isolated point which is far away from other points may not be clustered, and it won't affect the model of the cluster much.

5.3.3 DBSCAN

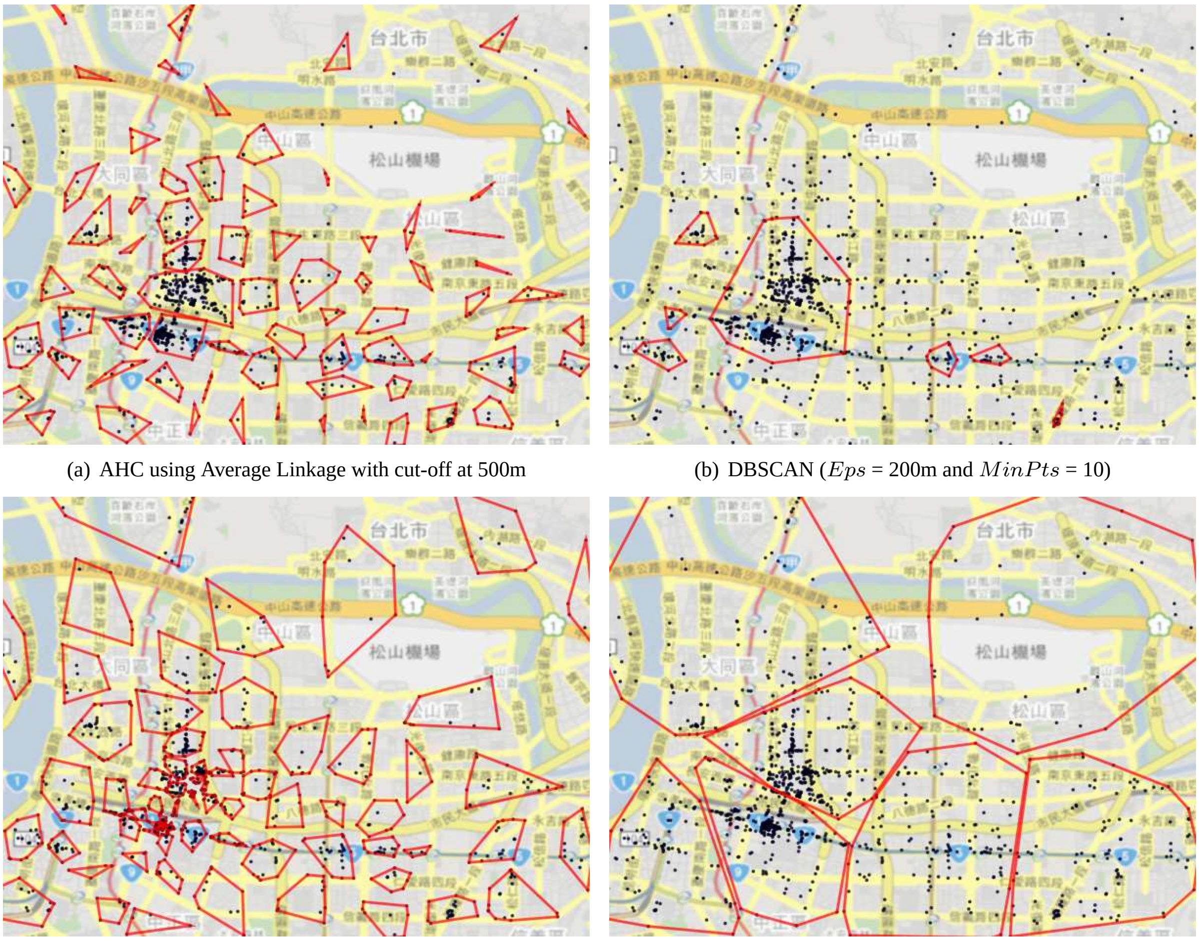
In DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [3], a spatial distance threshold Eps is used to define the proximity of two points. If the number of proximity of a specific point exceeds a predefined parameter $MinPts$, the point is regarded as in the core of one cluster, and its proximity belongs to the same cluster it is in. If the number of proximity is less than the parameter $MinPts$, the point may be at the border of one cluster, or an outlier to the population. This density-based algorithm deals with outliers and noises better than pure partitional clustering or hierarchical clustering.

5.3.4 Comparison

Four combinations of algorithms and parameters were executed and compared. The average linkage model was used for the agglomerative hierarchical clustering, and the cut-off distance was set to 500 meters; that is, the mutual distances between two records inside one cluster do not exceed 500 meters. Under this setting, 70 clusters were found in Fig. 7(a). The density threshold was set as 10 points in the radius of 200 meters for DBSCAN. Under this setting, records are grouped into 7 clusters in Fig. 7(b). For K-means, the number of clusters were choosen as 70 and 7, the same of other two algorithms. The results are shown in Fig. 7(c) and Fig. 7(d).

In addition to visualizing the clusters, quantitive comparison was provided. Five measurements were used: number of clusters, number of points per cluster, standard deviation with respect to the center of the cluster, area of the convex hull generated by the points in the cluster, and the density of the cluster (Fig. 8). In general, high-densed clusters with small standard deviation are preferred.

From the results, pros and cons of the algorithms are clear. K-means does not perform very well with small k value on large scale data with noises. The standard deviation of the points in the same cluster is the largest among four implementations. Large k gives better results than small k, but the standard deviation is still large. This disadvantage may come from the hard partition characteristic that every points should belong to one cluster. The property forces some clusters to absorb the noises and the standard deviation increases. The standard deviation of agglomerative hierarchical clustering is the smallest. However, the algorithm may generate small clusters with few elements. The smallest cluster generated in the experiment only contains 3 records. DBSCAN with proper parameters may treat these isolated points as outliers and ignore them in clustering; the clusters it generates quarantee a minimum number of elements. However, it may generate few clusters when facing a sparse dataset. It is not suitable at the beginning of time after the system gets online.



(a) AHC using Average Linkage with cut-off at 500m

(b) DBSCAN (Eps = 200m and $MinPts$ = 10)

(c) K-means (k = 70)

(d) K-means (k = 7)

Fig. 7. Clusters generated by the clustering algorithms.

6. Conclusion

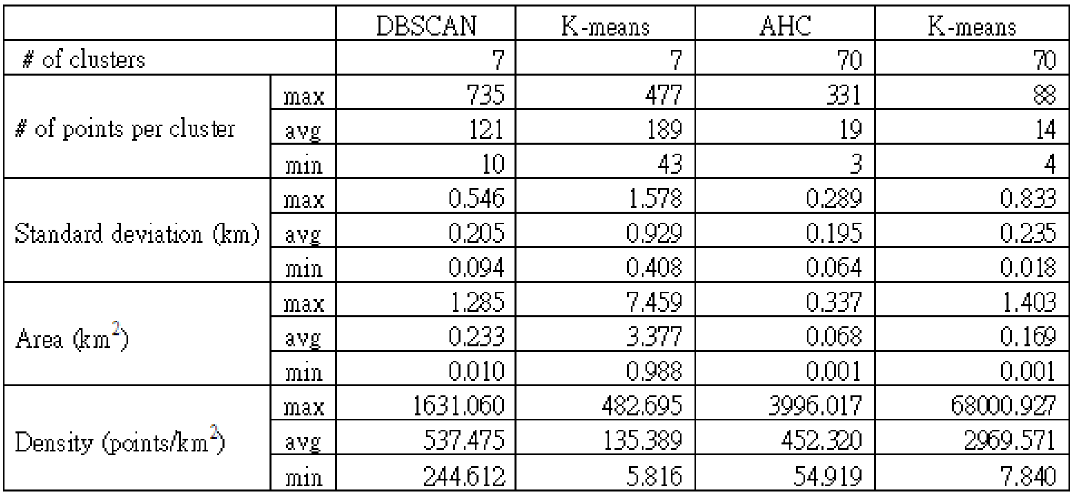
In this work, a four-step approach is proposed to solve the taxi demand analysis problem. Considering the context, taxi request records are filtered. These records are clustered according to the spatial distance. For each cluster identified, corresponding roads are found, and the cluster is associated to the semantic meaning of the representative roads. Hotness index is then calculated based on the property of the clusters and the distance from the taxi driver to the cluster. Combining all the information, context-aware association rules can be derived.

Different clustering methods have different performances on different kind of data distributions. In thie work, among the three algorithms applied, it is hard to take one as the best among the three. Hard partitional clustering like k-means is sensitive to outliers and noises. Agglomerative hierarchical clustering contains many unprevalent areas. Density-based algorithm like DBSCAN depends on two parameters and finding the proper parameters is not an easy task. It requires much more efforts to find an algorithm with the advantages of these clustering algorithms.

7. Future Work

In the context-based filtering process, the contexts may be relaxed, and the records under the relaxed contexts are considered for further computation. However, the relaxed contexts are not the same with the original one. The system should provide some mechanisms, such as adding discounts, to distinguish between the original records and records after relaxation.

After identifying the hotspots from large amounts of records, reasoning the causes is the next step. Location, time, and weather context are not enough to well explain the existence of the hotspots. It can be assumed that events may affect the distribution of taxi requests. The massive demand at the arena after the end of one famous musical show is an example. This event context can be retrieved from semantic web, and event ontology helps define the association between the event and the taxi demand. In addition, once the association rule is verified, future similar events according to the ontology can help predict the hotspots more accurately.



		DBSCAN	K-means	AHC	K-means
# of clusters		7	7	70	70
# of points per cluster	max	735	477	331	88
	avg	121	189	19	14
	min	10	43	3	4
Standard deviation (km)	max	0.546	1.578	0.289	0.833
	avg	0.205	0.929	0.195	0.235
	min	0.094	0.408	0.064	0.018
Area (km^2)	max	1.285	7.459	0.337	1.403
	avg	0.233	3.377	0.068	0.169
	min	0.010	0.988	0.001	0.001
Density (points/km^2)	max	1631.060	482.695	3996.017	68000.927
	avg	537.475	135.389	452.320	2969.571
	min	244.612	5.816	54.919	7.840

Fig. 8. Comparison of clustering algorithms on a total of 1,326 records.

Acknowledgements

This study is conducted under the “Intelligent Web-enabled Service Research and Development Project” of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China. Thanks to all the contributors and supporters to this project from collecting data, discussing ideas, implementing algorithms to reviewing the paper.

References

[1] T. Anderson, “Comparison of spatial methods for measuring road accident ‘hotspots’: a case study of London,” *Journal of Maps*, 2006, pp 55–63.

[2] D. Ashbrook and T. Starner, “Using GPS to learn significant locations and predict movement across multiple users,” *Personal and Ubiquitous Computing*, Vol. 7, no. 5, October 2003, pp 275–286.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996)*, Portland, Oregon, USA, August 1996, pp 226–231.

[4] N. Levine, “CrimeStat III: a spatial statistics program for the analysis of crime incident locations (version 3.0),” *Houston (TX): Ned Levine & Associates*, 2004, pp 290–387.

[5] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. I: Statistics, 1967, pp 281–297.

[6] F. Murtagh, “A survey of recent advances in hierarchical clustering algorithms,” *The Computer Journal*, Vol. 26, no. 4, November 1983, pp 354–359.

[7] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, “A clustering-based approach for discovering interesting places in trajectories,” in *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC)*, Fortaleza, Ceara, Brazil, March 2008, pp 863–868.

[8] D. Pelleg and A. W. Moore, “X-means: Extending K-means with efficient estimation of the number of clusters,” in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, San Francisco, CA, USA, June 2000, pp 727–734.

[9] G. Schwarz, “Estimating the dimension of a model,” *The Annals of Statistics*, Vol. 6, no. 2, March 1978, pp 461–464.

[10] S. Shekhar and Y. Huang, “Discovering spatial co-location patterns: A summary of results,” in *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD 2001)*, Redondo Beach, CA, USA, July 2001, pp 236–256.

[11] T. Vincenty, “Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations,” *Survey Review*, Vol. 22, no. 176, April 1975, pp 88–93.

[12] R. Xu and D. Wunsch II, “Survey of clustering algorithms,” *IEEE Transactions on Neural Networks*, Vol. 16, no. 3, May 2005, pp 645–678.