Context-aware taxi demand hotspots prediction

Han-wen Chang, Yu-chin Tai
and Jane Yung-jen Hsu*

Department of Computer Science and Information Engineering,
National Taiwan University, Taiwan
E-mail: r96005@csie.ntu.edu.tw
E-mail: r96047@csie.ntu.edu.tw
E-mail: yjhsu@csie.ntu.edu.tw
*Corresponding author

Abstract: In an urban area, the demand for taxis is not always matched up with the supply. This paper proposes mining historical data to predict demand distributions with respect to contexts of time, weather, and taxi location. The four-step process consists of data filtering, clustering, semantic annotation, and hotness calculation. The results of three clustering algorithms are compared and demonstrated in a web mash-up application to show that context-aware demand prediction can help improve the management of taxi fleets.

Keywords: hotspot mining; data mining; clustering.

Reference to this paper should be made as follows: Chang, H-w., Tai, Y-c. and Hsu, J.Y-j. (2010) 'Context-aware taxi demand hotspots prediction', *Int. J. Business Intelligence and Data Mining*, Vol. 5, No. 1, pp.3–18.

Biographical notes: Han-wen Chang is an MS student of Computer Science and Information Engineering at National Taiwan University since 2007. He is due to submit his thesis in July 2009. He received his BSc Degree from the Department of Computer Science and Information Engineering of National Taiwan University (2007).

Yu-chin Tai is an MS student of Computer Science and Information Engineering at National Taiwan University since 2007. He is due to submit his thesis in July 2009. He received his BSc Degree from the Department of Computer Science and Information Engineering of National Taiwan University (2007).

Jane Yung-jen Hsu is a Professor of Computer Science and Information Engineering at National Taiwan University. Her research interests include multi-agent systems, semantic data analysis, and service-oriented computing. She is actively involved in key international conferences as organisers and program committee members, and serves on the editorial board of the *International Journal of Service Oriented Computing and Applications*. She is a member of AAAI, ACM, IEEE, and TAAI.

1 Introduction

According to the Institute of Transportation (IOT) Survey of Taxi Operation Conditions in Taiwan Area 2006, in average, each taxi driver operated the business 9.9 h a day, driving approximately 147.3 km. However, about one-third of the time, 3.2 h, drivers were on the roads without taking passengers. The time and energy wasting phenomenon is more severe in Taipei urban area. Taipei City Department of Transportation reported that in over 60–73% of their operation hours, taxi drivers were 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 taxi drivers do not know where potential customers are, leaving them with no choice but to wander around the city. The goal of this research is to predict the areas with potential demand from contexts and past history.

To solve the problem, understanding and constructing the model of the taxi demand are important. Analysing the data on past history, including the time and location passengers got on taxis, provides clues to the demand distribution. Given the contexts of time, location, and weather, 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 scores can be calculated. Combining the cluster geometries, the semantic road names, and the hotness scores, hotspots are defined. As a consequence, drivers can adjust their strategies according to the demand distribution prediction.

The remainder of this paper is organised 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, Sections 5 and 6 describe the implementation of the clustering methods and the experiment results. Finally, concluding remarks and future directions are stated in Section 7.

2 Related work

According to Merriam-Webster Dictionary,[1] a hotspot is a place of more than usual interest, activity, or popularity. As in the application of taxi demand analysis, hotspots are the places of more than usual occurrence; that is, the places with high density of demand. With clustering techniques (Xu and Wunsch II, 2005) used for grouping similar items, hotspots could be identified from spatial data. The most common clustering methods used in hotspot analysis include k-means (MacQueen, 1967), x-means (Pelleg and Moore, 2000), hierarchical clustering (Murtagh, 1983), and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996). CrimeStat (Levine, 2004) is a spatial statistics programme which supports different algorithms including k-means, hierarchical clustering and kernel density estimate. The programme helps the police to visualise the location distribution of crime incidents and discover the hotspots. Anderson (2007) compared the results of several clustering methods on road accident data in London, and pointed out that there is no universal definition of hotspots of road accidents.

There are researches focusing on finding significant locations from GPS traces. Ashbrook and Starner (2003) use k-means-like iterative approach to cluster places into locations. Palma et al. (2008) propose a clustering method based on speed measurement to distinguish stops and moves in a single trajectory. Our work is different from the above in that GPS trace records have strong spatial-temporal continuity, but taxi request records are not. In addition, GPS traces are from individuals and more personalised, while taxi requests are with less personalised factor.

Spatial co-location pattern mining is to find the set of spatial features that are frequently located together in spatial proximity (Shekhar and Huang, 2001). To find co-location rule with high prevalence and high confidence, approaches similar to association rule mining are used. The problem of taxi demand hotspot prediction could be viewed as the co-location pattern mining among taxi stops and other spatial features. Our work is different from co-location pattern mining because spatial features are not the only context considered. Context of temporal features are involved to find more specific context-dependent patterns.

OptiTaxi[2] is a taxi management service provided by Correlation Systems Ltd. OptiTaxi predicts the demand for taxi services according to locations and time, attemping to maximise profits of the entire taxi fleet. However, the locations as the units of demand prediction are pre-defined and fixed in the OptiTaxi system. In our work, we adopt clustering techniques to dynamically generate the areas from the demand history.

The contribution of this work is an application to solve the context-aware pattern mining from taxi request records by adapting existing approaches from clustering. 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 involve 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. Figure 1 shows part of the time ontology used in this work.

The location dataset D_L stores m_p landmarks and m_r roads. Each landmark or road p_i in the dataset is defined with its associated name $name_i$, the geometry representation $geom_i$ within the given coordinate system, and the category cat_i it belongs to. The geometry representations of landmarks are defined as the representative points, while the roads are as line segments. On these locations, geometry relationship functions, such as `COVERS`, and processing functions, such

as DISTANCE, are defined. COVERS indicates whether one location fully covers the other location; DISTANCE calculates the shortest distance between two locations. With these properties, the hierarchical structure of these locations in the spatial domain can be established.

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

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

Figure 1 Part of the time ontology (see online version for colours)

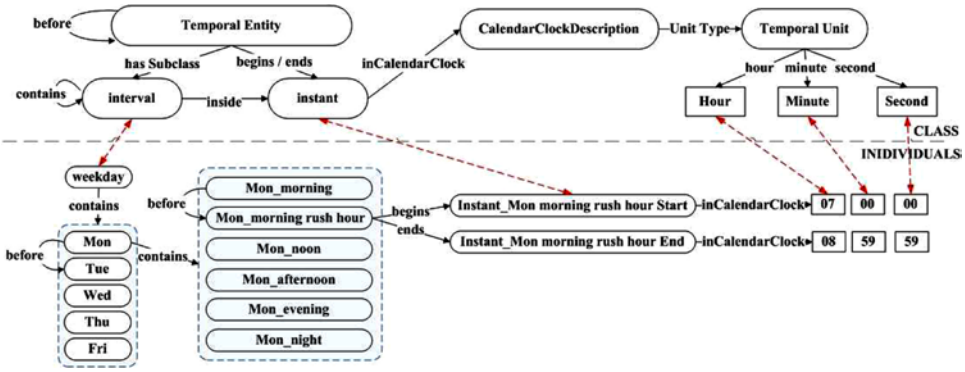
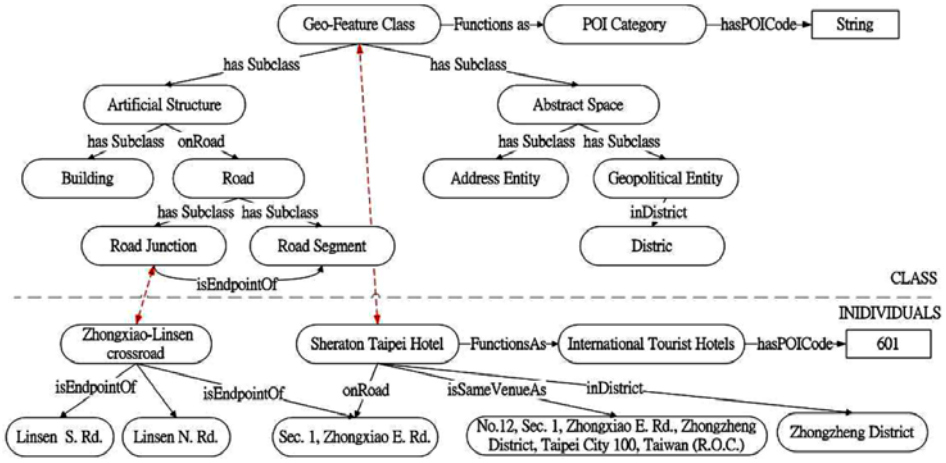


Figure 2 Part of the location ontology (see online version for colours)



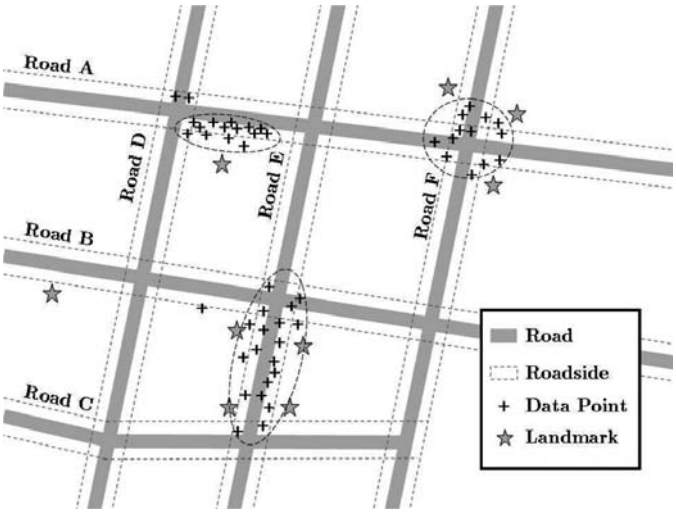
The request history dataset D_R stores n past taxi request records. A single taxi request record r_i contains the position including latitude ϕ_i and longitude λ_i, the timestamp $time_i$ when the customer gets on the taxi, and the weather condition w_i at that time.

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

The passengers making the requests may previously 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.

Take Figure 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) 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".

Figure 3 Roads, landmarks, and request records



When the system detects the need of prediction, such as when the latest passenger 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 characterised by the day of week d_t and the hour of the day h_t.

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

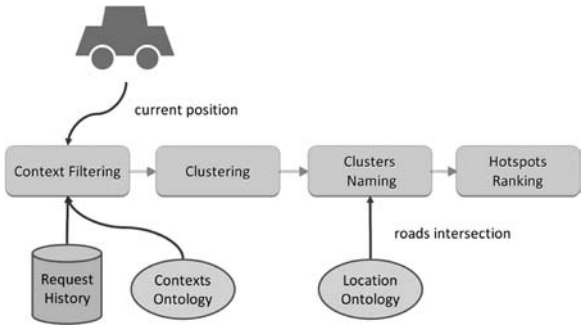
The expected output of the query $Query_t$ with the request history and location dataset D_R, D_L is a set of hotspots H which are composed of the geometries of clusters, the semantic names as roads or landmarks, and the corresponding scores $s_i \in [0..1]$ indicating the degree of hotness.

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

4 Proposed solution

Consider 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. Figure 4 shows the flow we used.

Figure 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

The context-based filtering routine is shown in Algorithm 1, which 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, which is from 7:00 AM to 9:00 AM. And all records under the relaxed context will be considered for the following steps.

Algorithm 1 ContextBasedFiltering(*requests*, *context*, *expected*)

inputs: *requests*, the set of past requests
context, the current context
expected, the expected size of requests required
returns: *filtered*, the set of filtered requests

```
filtered ← filter(requests, context)
n ← sizeof(filtered)
while n < expected do
    newcontext ← relax(context)
    if newcontext = context then
        return filtered
    end if
    context ← newcontext
    filtered ← filtered ∪ filter(requests, context)
    n ← sizeof(filtered)
end while
return filtered
```

4.2 Clustering positions

Clustering the GPS coordinates 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 analysing 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.

$$\mathrm{dist}(a,b) = 6372.795 \times \Delta\widehat{\sigma}(a,b)$$

$$\Delta\hat{\sigma}(a,b) = \arctan\left(\frac{\sqrt{(\cos\phi_b \sin\Delta\lambda)^2 + (\cos\phi_a \sin\phi_b - \sin\phi_a \cos\phi_b \cos\Delta\lambda)^2}}{\sin\phi_a \sin\phi_b + \cos\phi_a \cos\phi_b \cos\Delta\lambda}\right). \quad (1)$$

Clustering is the process to group similar items together, and the similarity measures of the items should be clearly defined before applying the clustering process. In spatial domain, distance measures are often considered as the similarity measure.

Suppose the two points a, b are at the positions $a = (\phi_a, \lambda_a)$ and $b = (\phi_b, \lambda_b)$ respectively. 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 (1975) formula is used with the assumption of spherical Earth. The radius of the Earth is assumed to be 6372.795 km, and the geodesic distance between two points is the radius times the angular distance $\Delta\widehat{\sigma}$ which is given in the equation (1).

The selection of similarity measure and clustering algorithm decides the result of clustering. There are several clustering algorithms nowadays, and each clustering algorithm 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 Section 5.

4.3 Mapping clusters to roads

Algorithm 2 outlines the steps for mapping clusters into the corresponding street names. 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, road junctions and road segments which match the clusters are identified, and the meanings of the clusters are assigned as the names of the junctions and segments.

Algorithm 2 ClusterNaming(*cluster*)

inputs: *cluster*, a set of locations
returns: *candidate*, the set of locations representing the cluster

1: $junctions \leftarrow$ FindJunctions (*cluster*)
2: $roads \leftarrow$ ConnectedRoads (*junctions*)
3: $candidate \leftarrow junctions \cup roads$
4: **return** *candidate*

Road junctions are the crossroads where roads intersect or connect. 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 hierarchical structure is defined in the ontology.

For each cluster, the system first locates the median position of the requests in the cluster and the nearest road junctions to this position. The Mean Squared Error (MSE) of the distance from this road junction to the request points is calculated as the threshold θ, and then all road junctions with MSE less than a portion of the threshold are listed (see Algorithm 3). Road segments whose two end points are in the junctions are also retrieved. The name of the junctions and road segments are considered as the semantic meaning of the cluster.

Algorithm 3 FindJunctions(*cluster*)

inputs: *cluster*, a set of locations
static: *junctions*, the set of road junctions
returns: *candidate*, the set of junctions representing the cluster

```
candidate ← {}
m ← median(cluster)
ĵ ← NearestJunction(m, junctions)
θ ← MSE(ĵ, cluster)
for all j_i in junctions do
    θ́ ← MSE(j_i, cluster)
    if θ́ ≤ k × θ then
        candidate ← candidate ∪ j_i
    end if
end for
return candidate
```

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 taxi driver, the prediction system should discover which area is the hottest. Two types of hotness scores are defined: the global hotness of cluster i, $\hat{s}_i$, and the personalised hotness in respect to driver j, $s_{i,j}$. The global hotness considers the size of the cluster area, A_i, the number of request points it includes, N_i, and the time span of the relaxed context of the requests, T_i. For example, if the records are from morning rush hour, 7 AM to 9 AM, during the two month period, the time span T_i is 120 h (two hours per day times 60 days). Intuitively, the cluster with more requests in the past is hotter; however, the number of requests is affected by the cluster size and the total time considered when retrieving the past data. As a result, the global hotness score is defined as the number of the requests divided by the size of cluster and the time span (see equation (2)).

$$\hat{s}_i = \frac{N_i}{A_i \times T_i}. \tag{2}$$

The personalised hotness score adjusts the global hotness score according to the distance from the driver to the cluster, D. The reasons to make adjustment are twofold. On the one hand, the nearby hotspots are preferred than the hot-but-far locations. Picking up the nearby customer reduces the time of driving without passenger, while travelling a long distance may cost more and have the risk that the customer is taken by another taxi. On the other hand, with personalised hotness scores, taxi drivers see the cluster differently, and the situation that all drivers rush to one particular place may reduce. As a result, the personalised hotness score is defined as the global hotness score divided by 1 plus the distance (equation (3)).

$$s_{i,j} = \frac{\hat{s}_i}{1 + D_{i,j}}. \tag{3}$$

5 Clustering methods

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.

5.1 K-means

K-means (MacQueen, 1967) 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 advantage of k-means resides in its ease of implementation and local minimal convergence. 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 dramatically to the result of k-means.

5.2 Agglomerative hierarchical clustering

Agglomerative hierarchical clustering approach (Murtagh, 1983) groups similar clusters/objects together one by one to form high-level clusters. 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 such 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 would not affect the model of the cluster much.

5.3 DBSCAN

In DBSCAN (Ester et al., 1996), a spatial distance threshold Eps is used to define the proximity of two points. If the number of proximity of a specific point exceeds the 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 partition-based clustering or hierarchical clustering.

6 Experiment result

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 are applied. The results are shown, and the comparison is provided.

6.1 Location dataset

The current location dataset is built based on the research data version 1.4 provided by the Institute of Transportation (IOT), MOTC, Taiwan.[3] 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.

6.2 Request data collection

The data collection process is supported by the Institute 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 makes 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, 2319 request records were collected. However, 487 records were ignored because their GPS readings are zeros or out of range.

The spatial distribution of the real taxi requests are shown in Figure 5. In this figure, clusters of points around Shandao Temple and Linsen North Road are obvious. The temporal distribution of each driver may be different (see Figure 6). Four out 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.

Figure 5 Spatial distributions of real requests

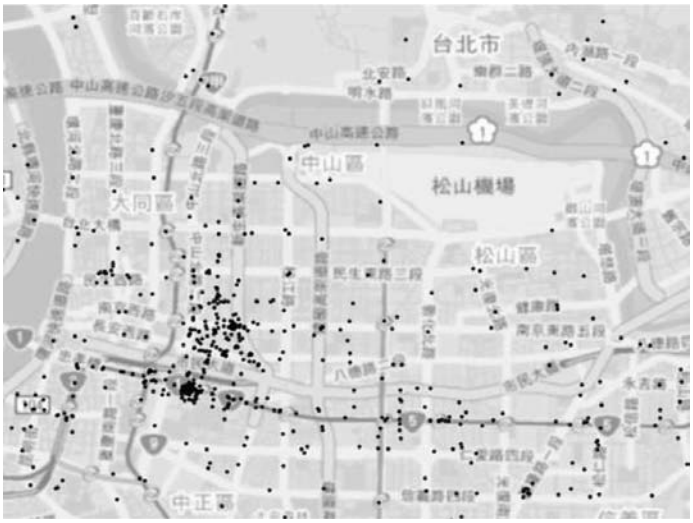
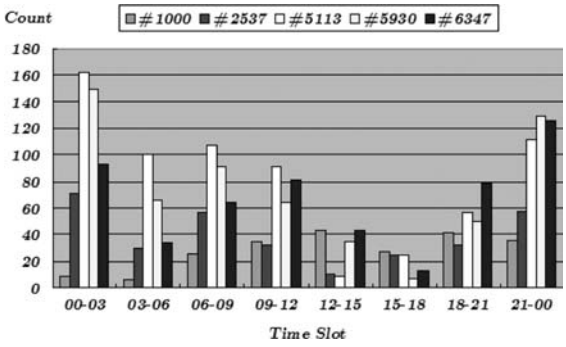


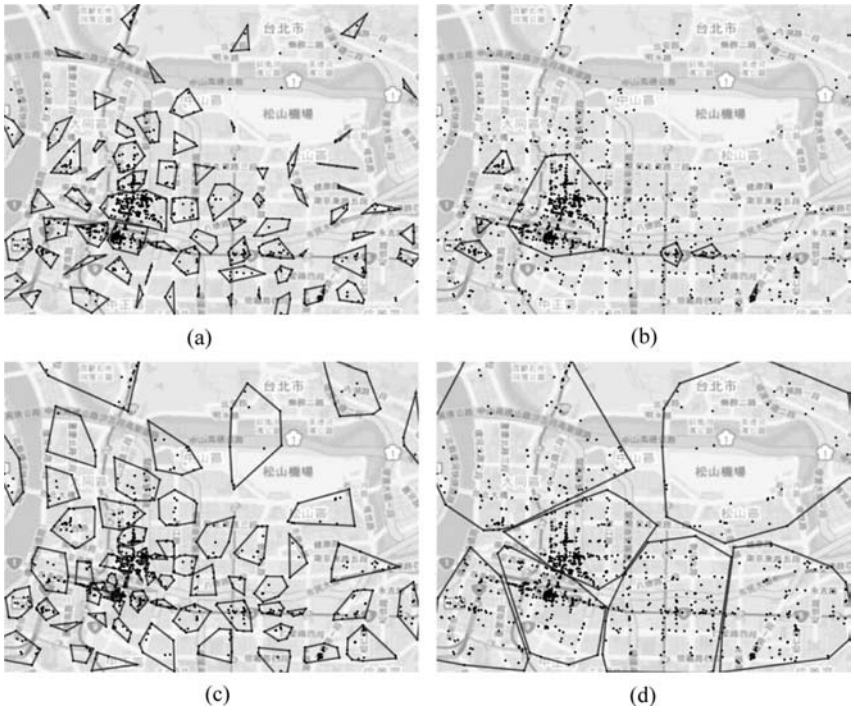
Figure 6 Temporal distribution group by drivers



6.3 Comparison of clustering algorithms

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 m; that is, the mutual distances between two records inside one cluster do not exceed 500 m. Under this setting, 70 clusters were found in Figure 7(a). The density threshold was set as 10 points in the radius of 200 m for DBSCAN. Under this setting, records are grouped into seven clusters in Figure 7(b). For K-means, the numbers of clusters were chosen as 70 and 7, the same of other two algorithms. The results are shown in Figure 7(c) and (d).

Figure 7 Clusters generated by the clustering algorithms: (a) AHC using average linkage with out of at 500 m; (b) DBSCAN ($Eps = 200$ m and $MinPts = 10$); (c) K-means ($k = 70$) and (d) K-means ($k = 7$)



In addition to visualising the clusters, quantitative comparison was provided. Five measurements were used: number of clusters, number of points per cluster, standard deviation with respect to the centre of the cluster, area of the convex hull generated by the points in the cluster, and the density of the cluster (Table 1). In general, clusters with high density and small standard deviation are preferred.

Table 1 Comparison of clustering algorithms on a total of 1326 records

		DBSCAN	*K-means*	*AHC*	*K-means*
No. of clusters		7	7	70	70
No. of points	max	735	477	331	88
per cluster	avg	121	189	19	14
	min	10	43	3	4
Standard deviation	max	0.548	1.578	0.289	0.833
(km)	avg	0.205	0.929	0.195	0.235
	min	0.094	0.408	0.064	0.018
Area	max	1.285	7.459	0.337	1.403
(km^2)	avg	0.233	3.377	0.068	0.169
	min	0.010	0.988	0.001	0.001
Density	max	1631.060	482.695	3996.017	68000.927
($points/km^2$)	avg	537.475	135.389	452.320	2969.571
	min	244.612	5.816	54.919	7.840

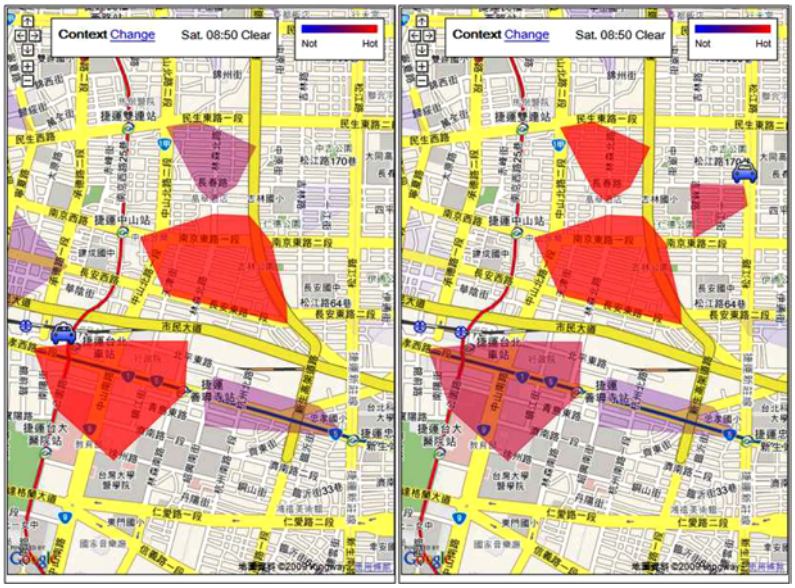
From the results, pros and cons of the algorithms are clear. K-means does not perform very well with small k value on large scale data with noises. The standard deviation of the points in the same cluster is the largest among four implementations. Large k gives better results than small k, but the standard deviation is still large. This disadvantage may come from the hard partition characteristic that every point 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 three records. DBSCAN with proper parameters may treat these isolated points as outliers and ignore them in clustering; the clusters it generates guarantee 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.

6.4 Mash-up application

A mash-up web application was create to demonstrate the hotspots prediction functionality. All contexts, including time, location, and weather, are retrieved from web services using Google AJAX API.[4] Google Maps is used to visualise the distribution of predicted hotspots, and a blue-to-red colour scheme is used to show the hotness score; the hotter, the redder colour. By clicking on the clusters, the descriptions of the cluster, in our work, the semantics in the form of name of roads and junctions, are shown in Figure 8. Figure 9 demonstrates the different views of two taxi drivers at the same context, which are the effects of personalised hotness score.

Figure 8 Demonstration of the mash-up application (see online version for colours)

Figure 9 Hotness scores are personalised according the drivers' locations (see online version for colours)



7 Conclusion and future work

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.

Different clustering methods have different performances on different kind of data distributions. In this work, among the three algorithms applied, it is hard to take one as the best among the three. Hard partition-based clustering like k-means is sensitive to outliers and noises. Agglomerative hierarchical clustering contains many irrelevant 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.

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 as 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.

The last but not the least, the 5-driver 2-month dataset cannot represent all taxi drivers for the whole year. Moreover, the dataset only records where drivers picked up customers, and there is no information about where drivers met no demands. To assess the effectiveness of the proposed solution, deployments of the proposed solution to a taxi fleet and a long-term evaluation on the average roaming time of taxis are necessary.

References

Ashbrook, D. and Starner, T. (2003) 'Using GPS to learn significant locations and predict movement across multiple users', *Personal and Ubiquitous Computing*, Vol. 7, No. 5, pp.275–286.

Anderson, T. (2007) 'Comparison of spatial methods for measuring road accident 'hotspots': a case study of London', *Journal of Maps*, Vol. v2007, pp.55–63.

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

Levine, N. (2004) *CrimeStat III: A Spatial Statistics Program for the Analysis of Crime Incident Locations (version 3.0)*, Ned Levine & Associates, Houston, TX, pp.290–387.

MacQueen, J. (1967) 'Some methods for classification and analysis of multivariate observations', *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. I, Statistics, University of California Press, Berkeley, CA, USA, pp.281–297.

Murtagh, F. (1983) 'A survey of recent advances in hierarchical clustering algorithms', *The Computer Journal*, Vol. 26, No. 4, pp.354–359.

Pelleg, D. and Moore, A.W. (2000) 'X-means: extending K-means with efficient estimation of the number of clusters', *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp.727–734.

Palma, A.T., Bogorny, V., Kuijpers, B. and Alvares, L.O. (2008) 'A clustering-based approach for discovering interesting places in trajectories', *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC)*, Fortaleza, Ceara, Brazil, pp.863–868.

Shekhar, S. and Huang, Y. (2001) 'Discovering spatial co-location patterns: a summary of results', *Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases (SSTD 2001)*, Springer-Verlag, Redondo Beach, CA, USA, pp.236–256.

Vincenty, T. (1975) 'Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations', *Survey Review*, Vol. 22, No. 176, pp.88–93.

Xu, R. and Wunsch II, D. (2005) 'Survey of clustering algorithms', *IEEE Transactions on Neural Networks*, Vol. 16, No. 3, pp.645–678.

Notes

[1]Source: http://www.merriam-webster.com/dictionary/hotspot
[2]http://www.optitaxi.com/
[3]Source: http://www.iot.gov.tw/english/ct.asp?xItem=187765&ctNode=2272
[4]Source: http://code.google.com/apis/ajax/