Detecting Meaningful Places and Predicting Locations Using Varied K-Means and Hidden Markov Model

Neelabh Pant* Ramez Elmasri†

Abstract

This work describes the use of Varied-K Means clustering and Hidden Markov Model techniques to predict a user's future movement based on the user's past historical data. Several techniques [1] [2] were proposed to predict a user's movement, but not many have concentrated on the user's location based on both weekday and time period within the day. We have introduced a method which models the user's data, not by just taking day of the week into consideration but also time interval. Our model is able to answer day-specific queries like *"Where is the user most likely to be when it is a Monday?"* or day and time-specific queries like *"Where is the user most likely to be between 6:00 pm and 9:00 pm on Saturdays?"* Our work gives us much higher prediction accuracy than previous research on this topic [9]. Such a model has multiple applications, which are described in the Introduction and Motivation section of the paper.

1 Introduction and Motivation.

In this paper, we have used real-world geospatial data (including latitude, longitude, day and time) recorded using GPS devices to extract meaningful locations, and model them in such a way that we may predict where a user will be at a given time and day of the week.

The dataset (GeoLife) contains GPS trajectories collected by Microsoft Research Asia from 178 users during a period of over 4 years. The dataset is a sequence of time stamped points containing latitude, longitude and altitude [13] [14] [15]. GPS loggers and GPS phones were used to record the trajectories approximately every 5 seconds. When a user's location change occured at a speed of less than 3 mph (assumed walking speed), the GPS logger did not record the data points. The dataset is distributed over 30 cities in China but mainly comprises locations in Beijing, China.The dataset was cleaned and uploaded to SQLite database.

Since the dataset is just a set of trajectories with a time difference of 5 seconds between pairs of consecutive points (latitude, longitude), there was a need to develop an algorithm which would intelligently group the points together to identify locations where a particular user spent most of their time. These locations are then considered as meaningful places where the user not only frequently went but also where the user spent substantial time.

Though the locations history of a user normally remains private and secured, if a user chose to share their locations with another individual, then an application of this research would be a shared location recommendation system. Consider a scenario where user1 and user2 have shared their locations histories with each other and are aware of each user's normal daily locations and tasks. This information can help them to plan tasks more efficiently and help each other if needed. For example, user1 has their **things to do** "input" list saved on their smartphone and "buying groceries" is listed as a pending task. The system predicts that user2 will be near or at a grocery store during the day while user1 is busy at the office and cannot go. With the help of such a system, user2 would be able to retrieve user1's grocery list and a reminder to buy groceries for user1.

One may also make use of this system in terms of online security. Any website where a user must log in to his/her account using their login credentials needs to place a special focus on securing the user's account on various devices. Such websites make use of the device's MAC address so that when a user tries to log in from any unknown device, the website requires the user to perform an extra step to confirm their identity. If such websites make use of location prediction technology, such as presented in this paper, then the website will automatically know the predicted behavior of the user and will recognize him/her without having to further confirm their identity.

2 Related Works.

The two-stage approach to location prediction has been used by several researchers. In [1] [2], the focus is on predicting user's location based on GPS data. They use K-means to cluster the locations, which influenced our approach, but they make use of a basic Markov Model

*University of Texas at Arlington. neelabh.pant@mavs.uta.edu

†University of Texas at Arlington. elmasri@cse.uta.edu

as part of their predictive model. Their prediction is dependent on the user's past location only and does not include day and time as in our work. We use Hidden Markov Model to incorporate weekdays and time as hidden states. Although a person's future location will depend on past and present locations, for a more comprehensive prediction we need to include weekday and time in the features set, which is lacking in [1] [2]. They can predict *"where someone will go next, but not when"*. In our research, we can answer the *"when?"* including weekday and time.

[12] has proposed a similar approach by using DBSCAN [6] instead of K-means [8] to cluster the data points, and using a variable order Markov Model instead of HMM for predictions. DBSCAN focuses mainly on the density of the data points and cluster them accordingly. Our K-means clustering algorithm can be customized based on different users and their pattern of stopping at specific locations, which is unrelated to the density of the data points. K-means gave us the liberty to set the number of K clusters, which was calculated by looking at the pattern of "time spent at locations" for each user. Hence, each user has different numbers of clusters based on their own behavioral pattern. Using Hidden Markov Model in our research we included weekday and time in our predictions. In our work, we are successfully able to answer the questions like *"Where is a user most likely to be at 7pm on Friday?"*. Such queries were not answered in [12].

[11] talks about predicting future locations based on historical data, but also address the *"data sparsity problem"*, which means *"unavailable historical trajectories"*. They propose a method which they call "sub-trajectory systhesis" (SubSyn) to address the data sparsity problem. They also have considered the privacy protection issue in order to hide sensitive location information of a user.

[9] proposed a different technique to cluster user's data based on temporal characeristics i.e. a cluster for sequences whose visits are made on weekdays by daytime (7am-7pm) and so on. The clustering algorithm is a very simple approximation that uses the timestamp of the last visited location. In our work, we have a more sophisticated algorithm that clusters a user's data points according to their pattern of stopping at a point of interest.

[10] on the other hand introduces a data mining approach to predict future location of a moving object. The author mines the database of moving object to match the unseen trajectory with the extracted trajectory to select the best association rule.

3 Clustering of Locations

We used the K-means algorithm to cluster the dataset but have modified it for our purpose. In general, the traditional K-means algorithm takes randomly-selected-user-defined k numbers of centroids to form k clusters. Each centroid compares its distance from the remaining set of points. The closest points to each centroid make a cluster, for which the mean location is calculated. The mean of all the points in a set is used as the new centroid for the next clustering iteration. This process is repeated until the mean stops changing. Once the mean no longer moves, all the points within it represent a cluster and are removed from consideration. This process is repeated until no centroid is remaining [8].

Our variation of the K-means algorithm was influenced by [1], [2], which concentrated on "*where the user is instead of how the user got there*". Our focus was to find locations where a user spent most of their time and to relate those locations to days of the week and times of the day. We targeted our algorithm to find the time elapsed between two consecutive points. If the elapsed time satisfied a threshold, then we marked it as a significant point according to that user. In this approach, we identified the points which have more than a threshold time difference "τ" between them and their corresponding previous points. Another challenge was to find a significant value of τ. In order to do this, we plotted a graph of *Graph to identify meaningful locations.*

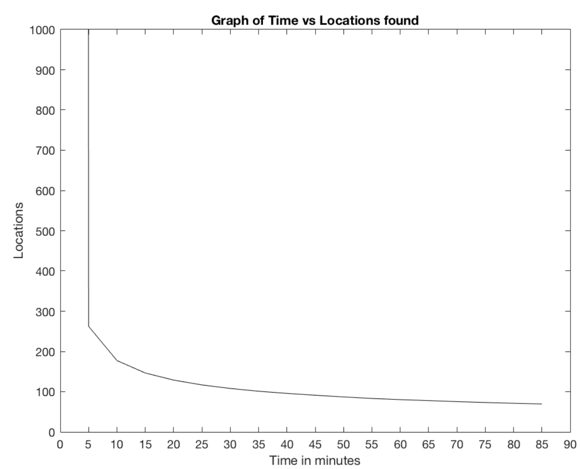

Figure 1: Graph to identify meaningful locations.

Figure 1 shows the graph that represents the average number of sites (*y-axis*) found for all 178 users when stopped for different durations of time (*x-axis*). We looked for the elbow or knee in the graph and made that duration of time a generalized condition for all the users on unique locations to focus on where each user stopped. As the time approached zero, the number of sites found were approximately 485,000.

After deciding on right time duration, which in our case was 10 minutes, we started extracting the sites' locations (latitude, longitude). Once we extracted all those points, we kept them in a set which we called the significant sites. In the traditional K-means algorithm, we would need to initialize the algorithm with a value of k. In our approach, the total count of significant sites where a user stopped for at least 10 minutes was chosen as the value of k, or the number of desired clusters.

The next step was to cluster points around these centroids. Since the data set consists of GPS recordings of people from China, the data is spread widely on a city-wide scale. We needed to have a good measure of the radius for a cluster. This was very important because if the radius was too large, we would end up with insignificant places in the cluster which will eventually give us incorrect results. If the radius was too small, we might end up getting one single point in the cluster. To select the best value of the radius for a cluster, we found the distances between each significant centroid. The distance between two points δ was calculated using the Haversine formula, which is defined as:

$$hav(\frac{d}{r}) = hav(\varphi_2 - \varphi_1) + cos(\varphi_1)cos(\varphi_2)hav(\lambda_1 - \lambda_2),$$

where hav is haversine function which is defined as:

$$hav(\theta) = sin^2(\frac{\theta}{2}) = (\frac{1 - cos\theta}{2}),$$

d = distance between two points,
r = radius of sphere,
φ_1, φ_2 = latitude of point 1 and point 2,
λ_1, λ_2 = longitude of point 1 and point 2, and
$\frac{d}{r}$ = central angle in radians.

We used the Haversine formula to calculate the great-circle distance between two points on a sphere from their latitudes and longitudes, which is the shortest distance over the earth's surface [3].

After calculating the distances between all sites, we extracted the minimum δ and used it as our radius of the clusters. For different users, δ came out to be different as one value of δ cannot be generalized for all the users. Figure 2 displays the clusters for a specific user, user3 in this case. The δ value was set to 0.2 miles.

The black dots are the clusters shown in the figure. There are a total of 253 clusters for user3, 6 of which are shown in the figure. These clusters, if examined closely, are around areas like a residential district, university (Tsinghua University, Beijing), hotel, airport, etc. A lot of information may be inferred from these clusters for better analysis, and we used such information for our prediction problem.

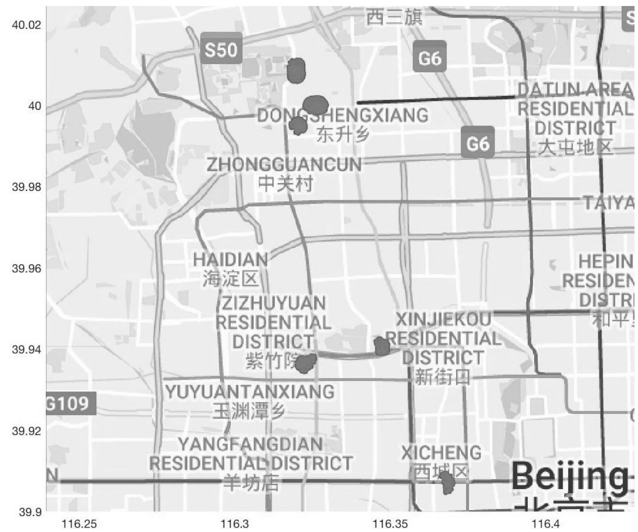

Figure 2: Clusters found for user3 when radius=0.2 miles

4 Hidden Markov Model

After we ran the clustering algorithm described in the previous section to create clusters of locations based on the two parameters time τ and radius δ, we got a set of clusters which represented the sites or locations where a user tended to visit. After we got all the clusters of locations, we updated our SQLite database where the GPS records are saved chronologically in order to update each user location with its respective cluster id (called LocID). One may name the clusters with specific names (if known) to make more sense of them, like "Home," "Grocery store," "Work place," etc., but we named them with integer ids for our purpose. The database table for a user consists of the records shown in table 1:

User_ID	Latitude	Longitude	DateTime	LocID

Table 1: Database Records of a User

Each latitude and longitude point is a member of a cluster and hence gets the cluster id in LocID attribute in the database table. After this process is completed, we have LocIDs in the database updated. These LocIDs are aligned chronologically, which helps us to get the transition from one cluster to another. As an example, figure 3 shows transitions between some of the sets of clusters for user3.

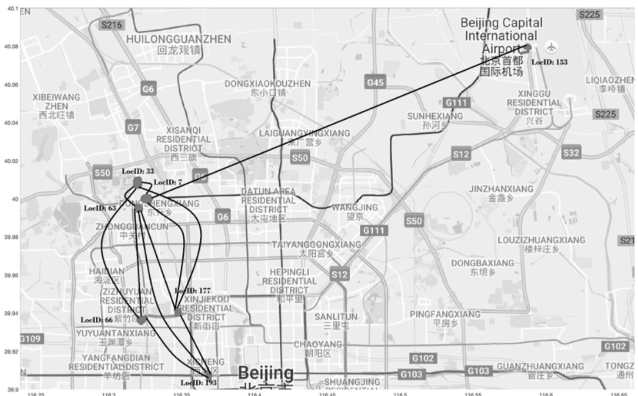

Figure 3: Transition Between Clusters

A line between any two clusters represents that the user moved from one cluster to another at some point in time. We do not show the direction of transition in figure 3, but direction is included in the data itself. We can thus analyze the number of times the user has traveled from one cluster to another. This will help us to calculate the probability of the user travelling from one cluster to another cluster.

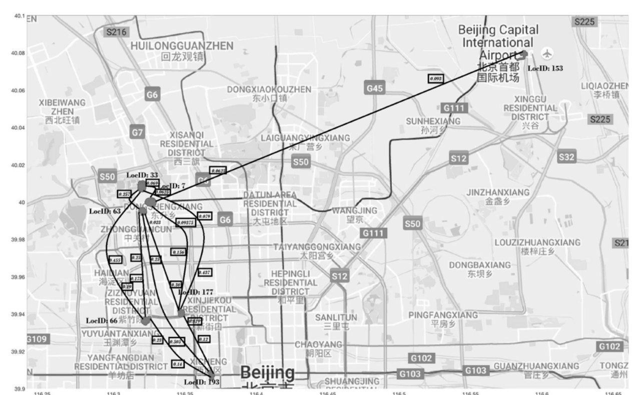

Figure 4: Transition Between Clusters with Probabiities

In figure 4, we have shown an example of how the transitions take place between different clusters. Figure 4 shows 7 clusters out of 212 clusters of user3. These clusters were given names using integer numbers (1, 2, 3, ..., 212). The ones which are shown in the figure are *LocID 33, LocID 7, LocID 63, LocID 66, LocID 177, LocID 193* and *LocID 153*. A markov model was created for each location in the map with transitions to other locations. Markov Models are state transition models with the nodes being the locations with corresponding state transitions between the nodes. It follows the Markov rule which states that the future state depends on the current state and observational data but are independent of past states [4]

Each cluster is a node in the Markov Model and the lines between each cluster represent the probability of the user to transition from one cluster to another cluster. For example, there is a 45% probability for user3 to go to *LocID 66* when the user is currently at *LocID 33* or 23% probability to go to *LocID 193* when the user is currently at *LocID 66*. We ran the algorithm on the whole dataset of user3, which gave us the results in form of transition probability which was used to answer queries like *"Where is user3 most likely to travel next if he is currently at location 46?*.

Places	Times	Transition	Frequency	Prob.
39	1	15 to 39	1/49	0.02
44	1	15 to 44	1/49	0.02
45	16	15 to 45	16/49	0.33
64	2	15 to 64	2/49	0.04
88	2	15 to 88	2/49	0.04
105	24	15 to 105	24/49	0.48
126	1	15 to 126	1/49	0.02
138	2	15 to 138	2/49	0.04
Total Visits	49			

Table 2: Probability of User3 from Location 15

Places	Times	Transition	Frequency	Prob.
7	1	67 to 7	1/44	0.02
66	31	67 to 66	31/44	0.7
100	3	67 to 100	3/44	0.07
108	1	67 to 108	1/44	0.02
182	2	67 to 182	1/22	0.04
194	2	67 to 194	2/44	0.04
212	4	67 to 212	1/11	0.09
Total Visits	44			

Table 3: Probability of User3 from Location 67

Such results are helpful to analyze the spatial behavior of a user, but with such analysis we miss out on the temporal behavior of the user. The primary objective of this research is to predict a user's location given a day and a time of the day through which we can answer queries like *"Where is a user most likely to be when it is a Monday?"* or *"Where is a user most likely to be when it is 6 pm on Sunday?"* or, *"Give me the next most probable location of a user when he is currently at home and it is Thursday at 5 pm."* Such queries cannot be obtained by the analysis done above. For such analysis, we make use of the Hidden Markov Model [5] where we introduce day and time as the hidden states for each visible state which, in our experiments, are the site locations.

There are multiple queries one may think of when predicting the location of a user. One of the most common prediction queries is to know where a user is on a specific day like Sunday, for example. To answer such a query, we make use of the Hidden Markov Model represented in figure 5, where A, B and C are the visible states, or the clusters, and the hidden states are the days of the week. Thus, by the use of Bayesian approach, we can answer the above query by

$$\mathbf{P}(\mathbf{x}|\mathbf{Sunday}) = \frac{\mathbf{P}(\mathbf{Sunday}|\mathbf{x}) * \mathbf{P}(\mathbf{x})}{\mathbf{P}(\mathbf{Sunday})}, \text{where}$$

$\mathbf{x}$ = ClusterID.

The query result is delivered by finding the $\mathbf{x}$ with the maximum probability. $\mathbf{P}(\mathbf{Sunday}|\mathbf{x})$, can be calculated by finding out the total number of visits to cluster $\mathbf{x}$ on **Sunday** divided by the total number of visits to cluster $\mathbf{x}$. $\mathbf{P}(\mathbf{x})$ is the ratio of total points in cluster $\mathbf{x}$ over the total number of points in all the clusters. Finally, if $\mathbf{X}$ = set of all clusters, then: $\mathbf{P}(\mathbf{Sunday}) = [\mathbf{P}(\mathbf{Sunday}|\mathbf{x}) * \mathbf{P}(\mathbf{x})] + [\mathbf{P}(\mathbf{Sunday}|\mathbf{y}) * \mathbf{P}(\mathbf{y})] + ... + [\mathbf{P}(\mathbf{Sunday}|\mathbf{n}) * \mathbf{P}(\mathbf{n})]$, where $(\mathbf{x}, \mathbf{y}, ..., \mathbf{n}) \in \mathbf{X}$.

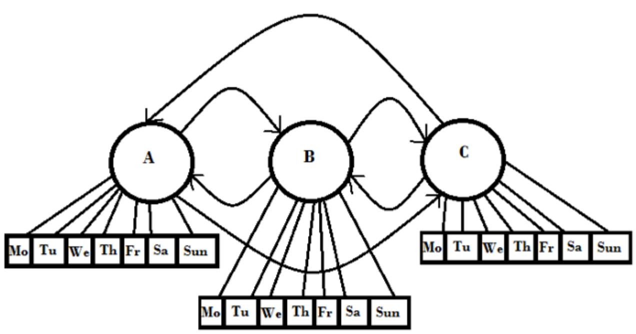

Figure 5: Hidden Markov Model for Days In a Week

Next, we take our model to one extra level where we let the model learn the pattern of a user not just based on days of the week but also including the time of day. With such a model, we would be able to answer queries like *"Where is a user most likely to be at 6 pm on Wednesday?"*. Figure 6 shows the model where the visible states remain the same as figure 5 but the hidden states have been expanded with the time of day. To reduce the number of hidden states, we divided the 24-hours of a day into 8 periods, each period consisting of a 3-hour interval, starting from 12am - 3am, 3am - 6am, ..., 9pm - 12am. In this way, we divided a day into 8 equal periods where each period of a day was a hidden state for each visible state. As an example, consider figure 6 below.

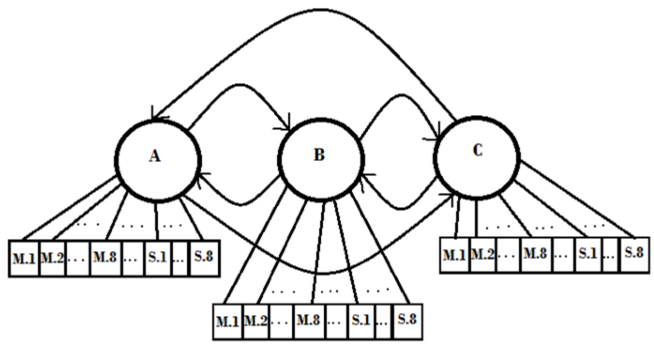

Figure 6: Hidden Markov Model for Days and Time

States A, B and C have their corresponding hidden states Monday 1st period, Monday 2nd period, ..., Sunday 7th period and Sunday 8th period. Each period will contribute towards the prediction by having some probability of its occurrence based on the user's historical data. For example, if we were to find the answer for the above query, which asks for the most probable location of a user when it is 6 pm on Wednesday, we will try to answer $\mathbf{P}(\mathbf{x}|\mathbf{Wednesday.6})$, which can be found by calculating:

$$\mathbf{P}(\mathbf{x}|\mathbf{Wednesday.6}) = \frac{\mathbf{P}(\mathbf{Wednesday.6}|\mathbf{x}) * \mathbf{P}(\mathbf{x})}{\mathbf{P}(\mathbf{Wednesday.6})}$$

Even though this looks like a simple conditional probability, if examined closely one may now see an extra feature in our calculations, the addition of time within the day. Intuitively, here we are first trying to calculate the contribution of quarter and day together for a user to be in a specific cluster. For simplicity, we calculate this beforehand by running the algorithm over the whole data set, which we will show in our experiment section, and later plugin the values for the final calculation.

5 Experiments

The system used to carry out the experiments had the following configuration: Processor: 2.2 GHz Intel Core i7, Memory: 16 GB, Programming language: MATLAB.

The predictive system we have built in our research is generalized for all the users and the experiments were conducted for 150 users with an average accuracy of 22%. Our accuracy was better than [9], which gave 13.85% prediction accuracy. We attribute this to including day of week and time interval in the HMM, and the use of varied K-means clustering. This is nearly a 70% improvement in prediction accuracy. In this paper we have shown the results for one specific user, i.e user3. user3 has about 500,000 data points.

The first set of experiments was to find the locations where a user is most likely to be on a specific day. Given a cluster $\mathbf{x}$ for user3, we use the following formula (Bayes theorem) to predict where user3 is on Sunday:

$$\mathbf{P}(\mathbf{x}|\mathbf{Sunday}) = \frac{\mathbf{P}(\mathbf{Sunday}|\mathbf{x}) * \mathbf{P}(\mathbf{x})}{\mathbf{P}(\mathbf{Sunday})}$$

We find the cluster $\mathbf{x}$ which has the maximum probability. For our experimental purposes, we used the data from multiple users whose data size ranged from 31,830 points to 935,576 points. For the purpose of this paper, we will show experiments for user3 only.

After, we found the clusters of user3, part of which is shown in table 2 and 3, we were interested to see where does this user often go on all the weekdays, i.e., Monday-Sunday. We ran the algorithm and following were the results (table 4). We show the top three clusters for each day of the week.

We then matched clusters with known locations on the map. We found places like the university "Tsinghua University, Beijing" and the housing area "Tsinghua Dormitory" and university departments like "Biomedical department" in cluster ID 208 (figure 7). This is why the probability of user3 being at cluster ID 208 is highest during the weekdays. The user may be a student, staff or faculty at the Tsinghua University and visits the university during weekdays. If we look at the probability distribution during the weekends, we see different cluster IDs like cluster 195 and cluster 85 having the highest probability.

In this experiment, we have taken a cluster's radius based on the δ value that was calculated in the previous section of clustering algorithm. Since the dataset is spread in a city-wide scale, and the δ came out to be 0.2 miles, we are predicting a user's location based on the clusters with size 0.2 miles. There could be another set of places within, say, cluster 208 where the user might be spending more time and which needs to be analyzed. To make more accurate predictions, we would need to cluster places within the clusters found above. This is one of the future works that we intend to work on.

Figure 8 displays cluster 195, which has the highest probability on Saturday. If examined closely, we find

Monday		**Tuesday**	
Cluster ID	Probability	Cluster ID	Probability
208	0.211	208	0.238
211	0.155	211	0.128
198	0.064	195	0.061
Wednesday		**Thursday**	
Cluster ID	Probability	Cluster ID	Probability
208	0.166	208	0.14
211	0.115	211	0.128
195	0.063	195	0.111
Friday		**Saturday**	
Cluster ID	Probability	Cluster ID	Probability
208	0.156	195	0.194
211	0.101	211	0.088
195	0.084	208	0.0825

Sunday	
Cluster ID	Probability
85	0.172
195	0.118
211	0.092

Table 4: Top Three Most Probable Locations With Day

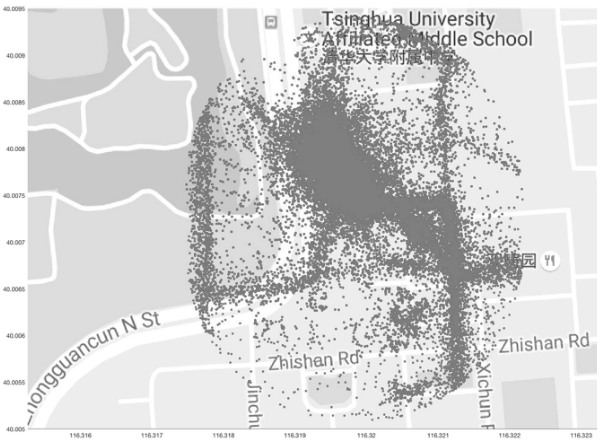

Figure 7: Cluster 208

places like hostels "PekingUni International Hostel" where a user could reside, food places where they might eat and places where a user could go in his/her free time.

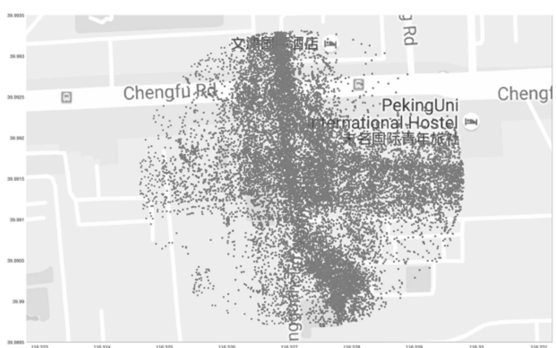

Figure 8: Cluster 195

In the second set of experiments, we were interested to find the answer to queries such as *"What is the most likely place for a user to be when it is 6 pm on Sunday?"* To answer this query, we made use of Hidden Markov Model as in figure 6. Each hidden state has its corresponding probability of occurrence with respect to the day of the week. We divided each day into 8 equal time intervals or periods. For example, Monday, 8th time interval, which is 9 pm-12 am, has a probability of occurrence for cluster ID, let's say, 208. In such a way, we calculated each interval's probability corresponding to its cluster along with probability of transition from one cluster to another. Once we calculated that, we constructed our Hidden Markov Model which was ready to answer queries like, *"What is the most probable location of a user when it is 2 pm on Monday?"* We ran this algorithm on user3 and found out locations where it is most likely for user3 to be at each period of the day (figure 6). Since the results generated were for each time interval and for each day of the week, for the purpose of this paper (instead of displaying all possible 56 results), in table 5, we have shown Monday's 1st, 2nd, 7th and 8th along with Sunday's 1st, 2nd, 7th and 8th period's results.

Monday.1		**Monday.2**	
Cluster ID	**Probability**	**Cluster ID**	**Probability**
208	0.18241	208	0.18241
211	0.11924	203	0.045506
203	0.045506	51	0.035027
...			
Monday.7		**Monday.8**	
Cluster ID	**Probability**	**Cluster ID**	**Probability**
208	0.18241	208	0.18241
211	0.11924	211	0.11924
195	0.075195	195	0.075195
...			
Sunday.1		**Sunday.2**	
Cluster ID	**Probability**	**Cluster ID**	**Probability**
208	0.18241	195	0.1936
211	0.11924	211	0.118
203	0.045506	85	0.0632
...			
Sunday.7		**Sunday.8**	
Cluster ID	**Probability**	**Cluster ID**	**Probability**
85	0.16245	85	0.15266
195	0.1265	195	0.11767
211	0.09879	211	0.03462

Table 5: Top Three Most Probable Locations With Day and Time

6 Future Works

The Hidden Markov Model is powerful enough to make good predictions, but lacks with respect to the change of behaviour or patterns of a user. For example, a user who is a student at a university enrolls to a set of classes in a semester. The student will follow the pattern set for that semester, which means that he will go to classes located in a building. But, once the semester is over, the student will totally change his pattern based on his new courses. This kind of change is difficult to adopt with the Hidden Markov Model and may cause inaccurate predictions. One possibility is to re-create the HMM for each semester. We also plan to introduce neural networks in our research such that it can be customized based on individual users. The neural network will adapt to the changes proactively and will make predictions based on just a given window size. We also plan to make changes to our current research by clustering locations within the wide radius clusters. This will help us to see the pattern of the user within a big cluster to make the analysis more complete.

7 Conclusion

In this work, we have demonstrated the use of the clustering algorithm K-Means and a predictive technology, the Hidden Markov Model, to predict a user's future locations. We introduced a method which will model the user's data not by just taking day of the week into consideration but also time interval within the day. Our model is able to answer day-specific queries like *"Where is the user most likely to be when it is Monday?"* or day and time specific queries like *"Where is the user most likely to be between 6:00 and 9:00 pm on Saturdays?"* Such a model can be applied to multiple applications which we discussed in the Introduction and Motivation section. We furthur plan to improve our work with the use of other new technologies which will benefit society in some form.

References

[1] D. Ashbrook and T. Starner, *Learning significant locations and predicting user movement with GPS*, Wearable Computers (ISWC). Sixth International Symposium on. IEEE, (2002).

[2] D. Ashbrook and T. Starner, *Using GPS to learn significant locations and predict movement across multiple users*, Personal and Ubiquitous Computing, 7(5) (2003), pp. 275–286.

[3] G. V. Brummelen *Heavenly mathematics: The forgotten art of spherical trigonometry*, Princeton University Press, 2013

[4] H. H. Bui *A general model for online probabilistic plan recognition*, IJCAI, Vol. 3:1309âĂŞ1315.

[5] S. R. Eddy *Profile hidden Markov models*, Bioinformatics (1998), 14(9), pp. 755–763.

[6] M. Ester, H. P. Kriegel, J. Sander and X. Xu. *A density-based algorithm for discovering clusters in large spatial databases with noise*, KDD (1996) Vol. 96, No. 34, pp. 226–231

[7] H. Gao, J. Tang and H. Liu. *Mobile location prediction in spatio-temporal context*, Nokia mobile data challenge workshop, (2012), Vol. 41, pp. 44

[8] J. A. Hartigan and M. A. Wong *Algorithm AS 136: A k-means clustering algorithm*, Journal of the Royal Statistical Society. Series C (Applied Statistics), (1979) 28(1), pp. 100–108.

[9] W. Mathew, R. Raposo and B. Martins *Predicting future locations with hidden Markov models*, ACM conference on ubiquitous computing, (2012), pp. 911–918.

[10] M. Morzy *Mining frequent trajectories of moving objects for location prediction, International Workshop on Machine Learning and Data Mining in Pattern Recognition, Springer Berlin Heidelberg. (2007), pp. 667–680*

[11] A. Y. Xue, R. Zhang, Y. Zheng, X. Xie, J. Huang and Z. Xu Destination prediction by sub-trajectory synthesis and privacy protection against such prediction*, ICDE, IEEE 29th International Conference, (2013), pp. 254–265*

[12] J. Yang, J. Xu, M. Xu, N. Zheng and Y. Chen Predicting next location using a variable order Markov model*, 5th ACM SIGSPATIAL International Workshop on GeoStreaming, (2014), pp. 37–42*

[13] Y. Zheng, Q. Li, Y. Chen, X. Xie and W. Y. Ma Understanding mobility based on GPS data*, ACM 10th international conference on Ubiquitous computing (2008), pp. 312–321.*

[14] Y. Zheng, L. Zhang, L. Xie and W. Y. Ma Mining interesting locations and travel sequences from GPS trajectories*, ACM 18th international conference on World wide web (2009), pp. 791–800.*

[15] Y. Zheng, X. Xie and W. Y. Ma GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory*, IEEE Data Eng. Bull. (2010), 33(2), pp. 32–39.*