2016-05-30

Semi-Autonomous Wheelchair Navigation With Statistical Context Prediction

Junqing Qiao
Worcester Polytechnic Institute

Semi-Autonomous Wheelchair Navigation With Statistical Context Prediction

by

Junqing Qiao

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Electrical and Computer Engineering

by

May 2016

APPROVED:

Professor Padir Taskin, Major Thesis Advisor

Professor Massoud Yehia, Head of Department

Abstract

This research introduces the structure and elements of the system used to predict the user's interested location. The combination of DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm and GMM (Gaussian Mixture Model) algorithm is used to find locations where the user usually visits. In addition, the testing result of applying other clustering algorithms such as Gaussian Mixture model, Density Based clustering algorithm and K-means clustering algorithm on actual data are also shown as comparison. With having the knowledge of locations where the user usually visits, Discrete Bayesian Network is generated from the user's time-sequence location data. Combining the Bayesian Network, the user's current location and the time when the user left the other locations, the user's interested location can be predicted.

Acknowledgments

First and foremost I would like to thank my advisor, Taskin Padir, for giving me this chance of doing this research and guiding me through this whole process. Even though I had an elementary understanding of machine learning, he still trusted me and gave me help when I needed.

I would also want to thank my friend and my coworker, Dmitry Sinyukov. All ideas in this whole project are the results of cooperation between Dmitry Sinyukov and me. I learned lots of useful techniques and methods from him.

I owe special thanks to my committee members, Lifeng Lai and Xinming Huang, for their useful and constructive comments and advice. They are not only the my committee members but also my respectful teachers. I learned a lot from them.

Last but not the least, I would like to thank my parents. I thank them for their not seeking return of love and their trust in me. They support me without any doubts on my decisions. They worked hard to pay my tuition fee. What they want is only that I can have a wonderful life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Electrical wheelchairs are widely used all around the world. Most electrical wheelchairs are controlled by joysticks. However certain groups of wheelchair users are unable or have difficulty to use this kind of interface. So wheelchair with a special control system is designed in [11] and [10] (See Figure 1.1). On this wheelchair, Laser range finder (LIDAR) and wheel-on-wheel encoder modules are installed for generating map of the surrounding environment and localizing. Additionally, ultrasonic and infrared ranges sensors are installed for obstacle avoidance. With those sensors the wheelchair can have the ability of navigating and avoiding obstacles. The only input needed from the user will be the destination. So in this research we studied how to find those possible destinations from the history location data of the user and estimate the probability of those destinations at a given time.

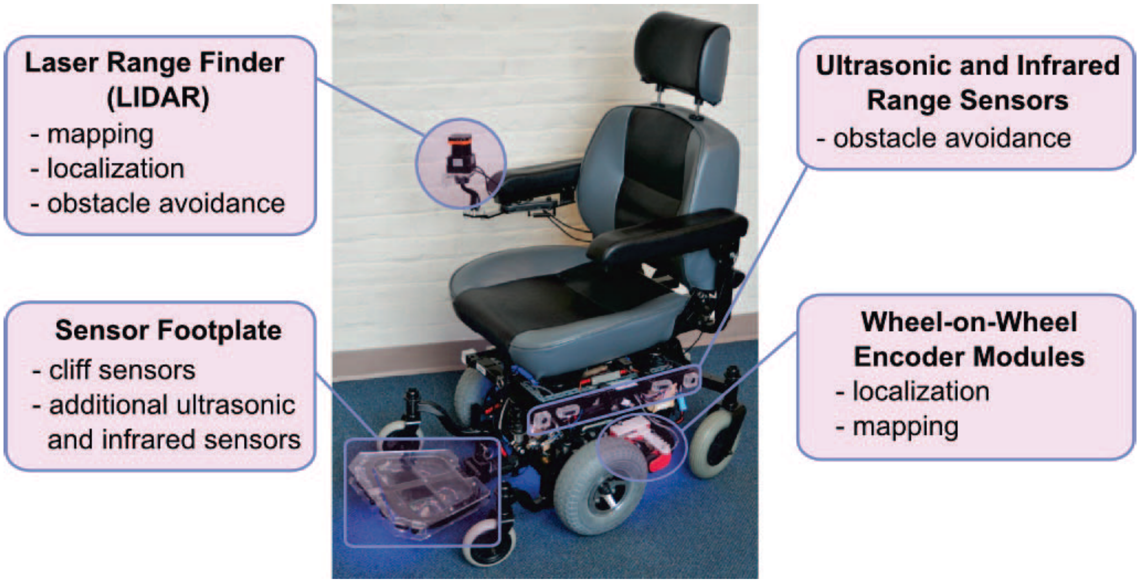

Figure 1.1: Photo of Robotic Wheelchair with Assistive Navigation[11]

1.2 Importance

With the development of sensors and robotics technology, robots are more and more widely used in home servicing. The algorithm of predicting the user's future location not only can be used in the wheelchair but also can be used in home servicing robots. Knowing the user's potential location at a given time may make the navigation system of robots more useful and human friendly. Such as, robots will get the knowledge about when the user has lunch, when the user goes back from work and when the user sleeps today from knowing the potential location at any given time. With such knowledge, smart cookers in the future can prepare food for the user at the right time. Moreover, smart home can turn the light on right before user arrive home from work. Likewise, cleaning robots can know the best time to clean the room without disturbing the user by knowing when the user will be out of the room.

1.3 Thesis Goals and Overview

The goal of this thesis is

> *To develop a technology to learn the relationship between locations and time from the location data of the user, as well as predict the location of the user ahead of time.*

The remainder of the thesis document is structured as follows:
In Chapter 2, a short review of algorithms for learning the habit of users is presented.
In Chapter 3, the overall methodology of this user location prediction system is introduced.
In Chapter 4, the clustering algorithm used to divide each location automatically is described. At the same time some background on clustering algorithms will be provided.
In Chapter 5, a detailed description of how to learn the relationship between those location clusters gathered in chapter 3 is described. There will be two parts in this chapter. The first part is a short introduction of Bayesian Network and the method of learning Bayesian Network automatically. The second part presents the work of learning discrete Bayesian Network from the gathered data.
Finally, in Chapter 6 I conclude the whole thesis and discuss about the future work.

Chapter 2

Literature Review

With the recent development in sensor system and embedded technology, it has become possible to collect location and behavior information of people in daily lives[5]. So there are more and more researches and application about learning and predicting the behavior of people. In the following paragraphs, several typical solutions and algorithms on this field will be introduced.

2.1 Time-series Data Mining

[5] is aimed on this subject. The authors introduced a prediction system for supporting users' daily lives. They use embedded sensors to record the behaviors of users in daily-life as the input to the prediction system. The prediction system learns the characteristic patterns of the user's behavior. The method applied in that paper discovers time-series association rules, which is a frequent combinations of events called episodes. And finally, the prediction system can output the prediction of the future behaviors based on the rules and the behaviors which are observed currently by the sensors.

The authors discussed three methods which could express time-series data. They are Time-Series Data Comparison Method by Chiu et al [1], Hidden Markov model [8] and Time-Series Data Mining Method by Das et al [3]. Those methods are used to find relationships between events. The first method is possible to add weight to transitional parts in the time-series data. However, the data which can be used in the first method is limited to one dimensional, so it has difficulty to express information from multiple sensors. The second method can deal with the transition of multidimensional data. But it is hard to treat data where the value rarely changes. The third method can convert time-series information into a set of events. Then rules can be found among those events. After comparing those methods, the authors selected time-series data mining method as the base method. To predict one behavior, at first, lists of pre-behavior episodes are generated from data gathered by sensors. Secondly, the pre-behavior episode occurrence rate is calculated. Pre-behavior episode occurrence rate is the frequency of the specific episode occurring before the behavior of interest happening. The rate is equal to how likely the predicted behavior will happen given the pre-behavior episode. Thirdly, authors user J measure from [12] to evaluation the behavior prediction rules. J measure is a value that shows how much ambiguity is removed in the phenomenon if a specific rule is given. A larger J measure means the rule gives more accurate information[5]. Finally, prediction can be made based on rules. However, during our research, we find that some behaviors of people are based on the time of the day instead of the events happened before. It is explain in the Method Selection section of Chapter 5.

Table 2.1[1] shows top 10 rules generated by the algorithm mentioned above. Those 10 rules are all used to predict the same behavior of the user. In the episode column of this table, every letter indicates a sensor. The center mark of each letter is the state of the sensor. The Accuracy is when the sensors' states are identical to the

Rank	Episode	Accuracy(%)	J-measure ($\times 10^{-4}$)
1	$D_{1,on}$	26	9.74
2	$F_{5,on}$	9	7.74
3	$F_{5,on}, D_{1,on}$	28	7.65
4	$F_{4,on}, D_{1,on}$	58	5.18
5	$F_{2,off}, D_{1,on}$	92	4.60
6	$F_{2,off}, F_{5,on}, D_{1,on}$	92	4.60
7	$F_{4,on}$	12	4.56
8	$F_{5,on}, F_{4,on}, D_{1,on}$	69	4.54
9	$F_{2,off}, F_{5,on}$	73	4.30
10	$F_{2,on}, D_{1,on}$	92	4.17

Table 2.1: Top 10 rules found in[1]

episode, the frequent of the behavior of interesting happens. We can find that for some rule the result is good. But for the other rules the result is not so good. What's more, the authors only focused on events happened before the behavior of interest. However, some behaviors of people are based on the time of the day instead of the events happened before. This system may not have good performance on behaviors strictly related to the time of the day such as having breakfast or watching specific TV show.

2.2 Dynamic Bayesian Network

[7] is also very closed to this filed. The authors use dynamic Bayesian network for both navigation and goal prediction. The Figure 2.1 shows the structure of the dynamic Bayesian network used. (a) depict the dynamic Bayesian network of the topological navigation system. (b) is the combination of destination prediction

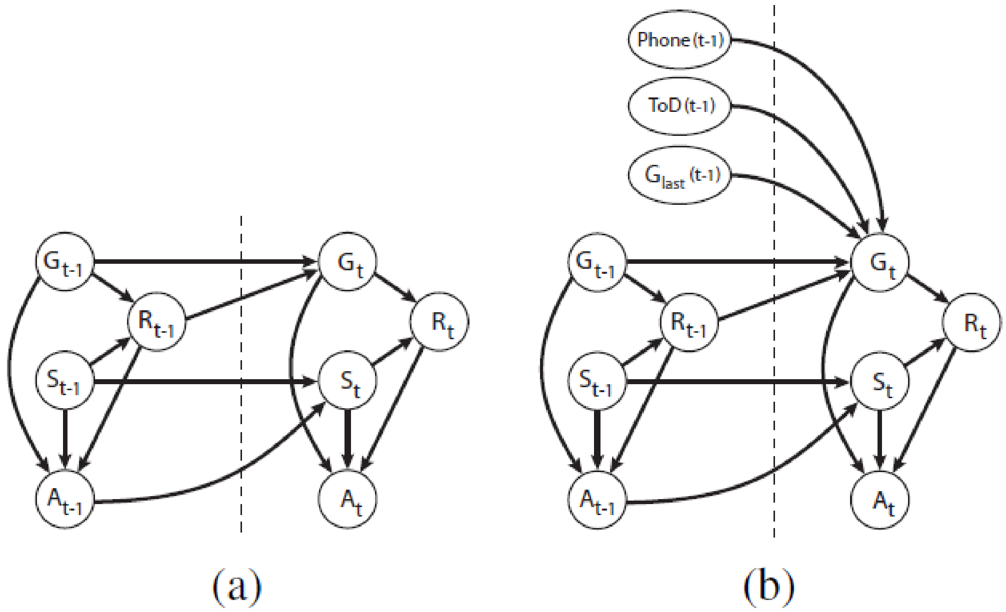

Figure 2.1: Dynamic Bayesian network of [7]

and the navigation system. And S is States, A is Actions, G is Goals, R indicates whether goal is reached, G_{last} is the last visited goal and ToD is Time of the Day. Our discussion will focus on the goal prediction part. At first, this system discretizes the day into several time intervals, such as waking up, lunch, or evening by using the node ToD, time of the day. And inside each time intervals, it maintains a probability table to save the probability of $P(G_t|G_{t-1}\ R_{t-1}\ G_{last}\ ToD\ Phone)$. G_{t-1} is the last goal, R_{t-1} indicates whether goal is reached, G_{last} is the last goal, ToD is the time of the day and $Phone$ means whether the phone is rining or not. In the experiment performed in [7], this system can predict goals correctly in 85% of the time.

In next chapter, the methods used to achieve the goal of this research will be presented.

Chapter 3

Methodology

The method used in this destination prediction system in introduced in this chapter by the flow chart in Figure 3.1. Inside the flow chart, the rounded rectangle represents the wheelchair with physical sensors embedded. Rectangle nodes represents processes. In addition, parallelogram nodes are input or output files used by each process.

At first, location and speed data will be collected from the wheelchair. There are two modules to finish this task. One is Data Collector which is used to collect the location and direction data from the wheelchair and save all the data into a file continuously. This file records all location and direction data in time order. By doing this the wheelchair can have enough data to learn the relationship between locations and time. The other module is State Observer which is used to observe the current state of the wheelchair. The current state includes the current position of the wheelchair and the times when each location was visited last. This module will be used when probability map needs to be generated.

Secondly, data from the wheelchair will be analyzed. This task requires two steps. The first step is clustering. In this step, we divide the location data into several

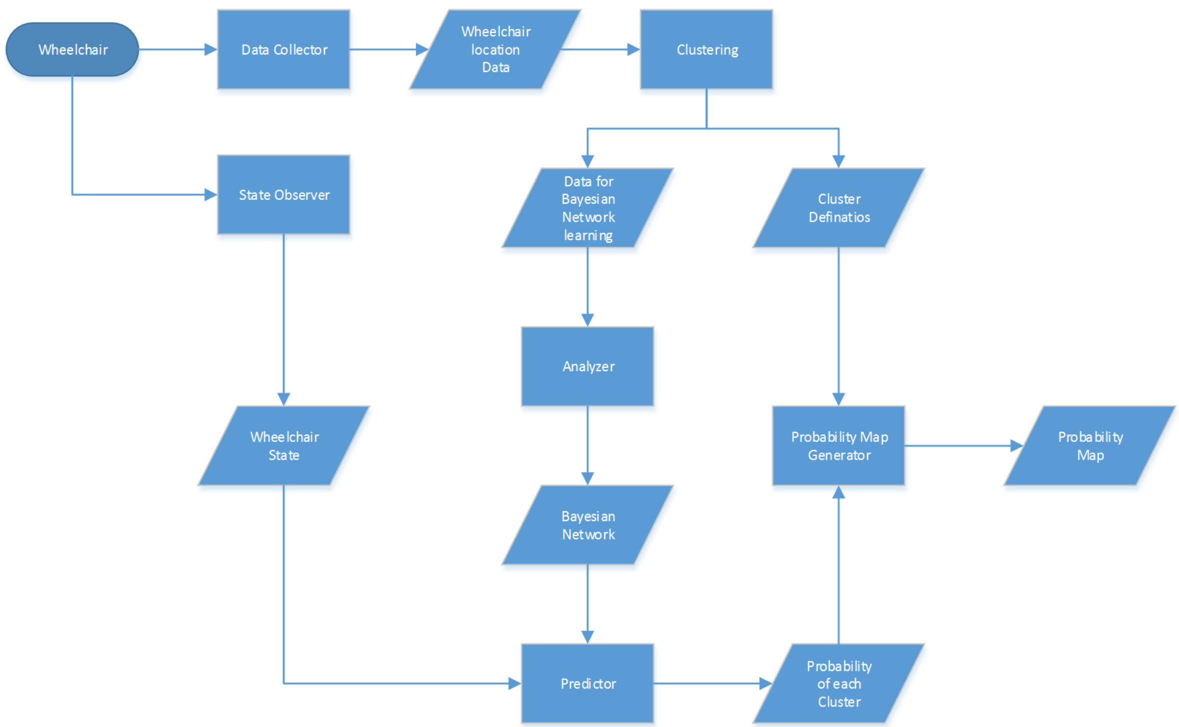

Figure 3.1: Flow chart of the destination prediction system.

clusters by performing the combination of density based clustering algorithm and the Gaussian mixture model. In the next step, we can just deal with those clusters instead of the massive location data. The second step is called Analyzer the aim of which is to analyze the relationship between those location clusters with respect to time. We use the Bayesian Network as the basic method for analyzing. After those two steps discussed above, a Bayesian Network connecting each location cluster will be learned.

Finally, the probability distribution map of the user's future location can be generated based on the wheelchair's state and the Bayesian network we get before. There are two modules inside this part. First one is the predictor which use the Bayesian network learned before and wheelchair's state as input. The output of this module is the probability of each cluster. The second part is Probability Map Generator which combines the probability of each cluster and the parameters of each cluster. The Probability map generator module can output the final result, the probability map of the user's further location.

Chapter 4

Clustering Location Data

In this chapter, clustering algorithm used to find locations where the user usually visit is presented. The location data used for this research are generated by Dimtry Sinyukov, a PhD student in WPI. He seated on this wheelchair for a whole week. So we can have the daily location data of a human user. This experiment was carry out at the first floor of the ECE department building of WPI. The map captured by Laser Range Finder(LIDAR) system is shown in Figure Wheelchair's location was gathered from Laser Range Finder(LIDAR) and wheel-on-wheel Encoder Modules on the wheelchair[11]. The Data Collector module recorded one entry of data every 20 seconds. Because the aim of this research to predict destinations of the wheelchair, we only cares about those stationary or moving slowly point. So, points whose speed are larger than certain threshold was removed. After this there are still thousands of entries of location points for only one day. To learn the relationship between locations and time from those data directly seems impossible. Solution for this problem is to treat locations near each other as the same location.

So the first step is to divide those data in to clusters based on location. Mainly, there are three reasons for the necessary of clustering. Firstly, if we process all

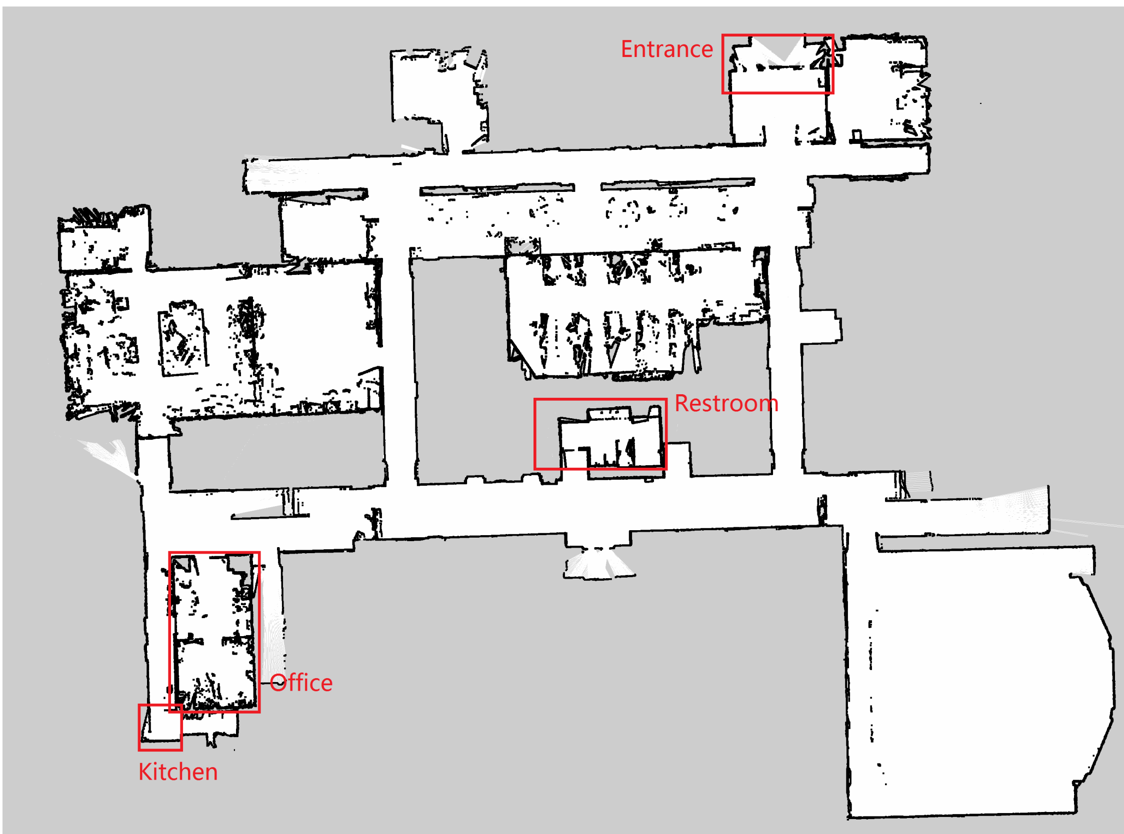

Figure 4.1: Experiment map gathered by Laser Range Finder(LIDAR) system

those entry of data individually, there will be too much data to be processed. As a result, the computing resources may not be enough. Secondly, clustering algorithm can get those regions where the user usually visit. Relationship between those regions can give us clues about the user's distention. In the following sections, I will discuss several popular clustering algorithms. For each clustering algorithm, short introduction will be given first. Then follows the testing result by applying each algorithm on real data gathered from the wheelchair.

4.1 Selection of Clustering Algorithms

The main responsibility of the clustering algorithm used on the wheelchair is to find all these regions where the user usually visit as well as to get the probability density distribution within those regions. The later is the conditional probability density distribution over all location points within the region. For each location point, this probability means how likely the point will be visited under the condition that the region is visited. It is hard to find the appropriate clustering algorithm without testing it on the data. So we tested 4 popular clustering algorithms. They are Connectivity-based clustering algorithm, Centroid-based clustering algorithm, Distribution-based clustering algorithm and Density-based clustering algorithm. The following subsections present how those clustering algorithm work and the testing result of applying those algorithm on data gathered from wheelchair by answering this 3 questions:

1. How does the clustering algorithm work?
2. What kind of data can be handled by the algorithm?
3. How well does this cluster algorithm work in our project?

4.1.1 Connectivity based clustering algorithm (also known as hierarchical clustering)

How does it work?

Hierarchical clustering algorithm is based on the idea that objects are more related to nearby objects than objects farther away. This algorithm has following 3 steps:

1. Find the similarity or dissimilarity between every pair of objects in the data set, in another word, finding the distance between each data point.

2. Group the objects into a binary, hierarchical cluster tree.

3. Cut the hierarchical tree into clusters.

Let's look at the 5 example points shown in Figure 4.2. First step is to find the distance between each point. Then, this algorithm combine every nearest two points into one cluster. In our case, point 1 and point 3 are combined into one cluster, point 4 and point 5 are combined into another cluster, as shown in Figure 4.3. This algorithm keeps combining every nearest two points or clusters into one new cluster until everyone is combined into one cluster. The final step is to select the level where the hierarchical cluster tree should be cut, as shown in Figure 4.4. The selection is usually based the desired number of clusters or the largest distance between points within one cluster.

What kind of data can be handled by Connectivity based clustering algorithm?

There are two requirements for this clustering algorithm: At first, the size, in another words, largest distance between two points in each cluster should be approximately same. Secondly, either the desired size of the cluster or the desired

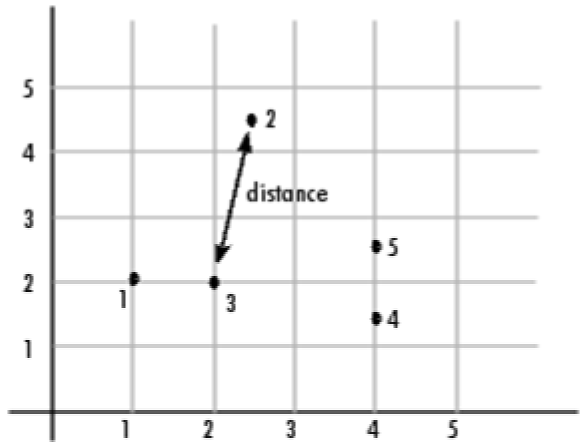

Figure 4.2: Demonstration for Hierarchical clustering from wikipedia

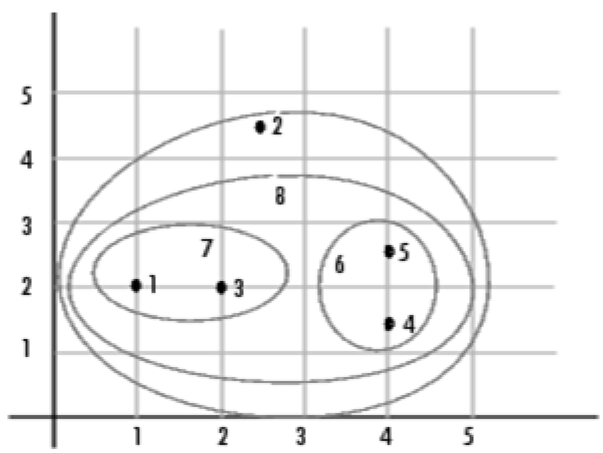

Figure 4.3: Clustering points into a hierarchical tree (from wikipedia)

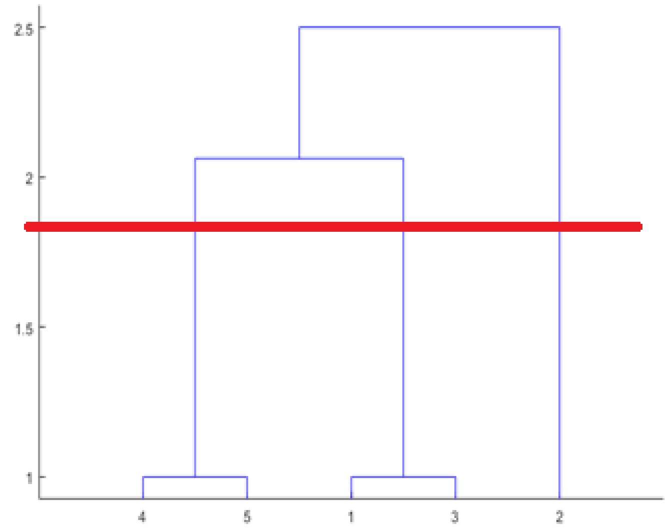

Figure 4.4: Cutting the cluster tree to get three clusters (from wikipedia)

number of clusters should be known.

The reason for the first requirement is the last step of the hierarchical clustering algorithm is to cut the hierarchical tree into clusters. Clusters at the same level of the hierarchical tree have the same size. And for the second requirement, the reason is that this algorithm have to cut the tree base on some stop criteria. And the desired size of the cluster or the desired number of clusters are two most common criteria.

Those two requirements indicate that this algorithm is hard to be used in our project. If we want to use this algorithm, we have to develop an algorithm to find the either the appropriate size fo the cluster or the number of all clusters.

Testing Connectivity based clustering algorithm with user's location data:

To use this hierarchical clustering algorithm, the number of clusters should be given in advance. So we tried several numbers. We found that setting the number of clusters to be 10 had the best result which is shown in Figure 4.5. However,

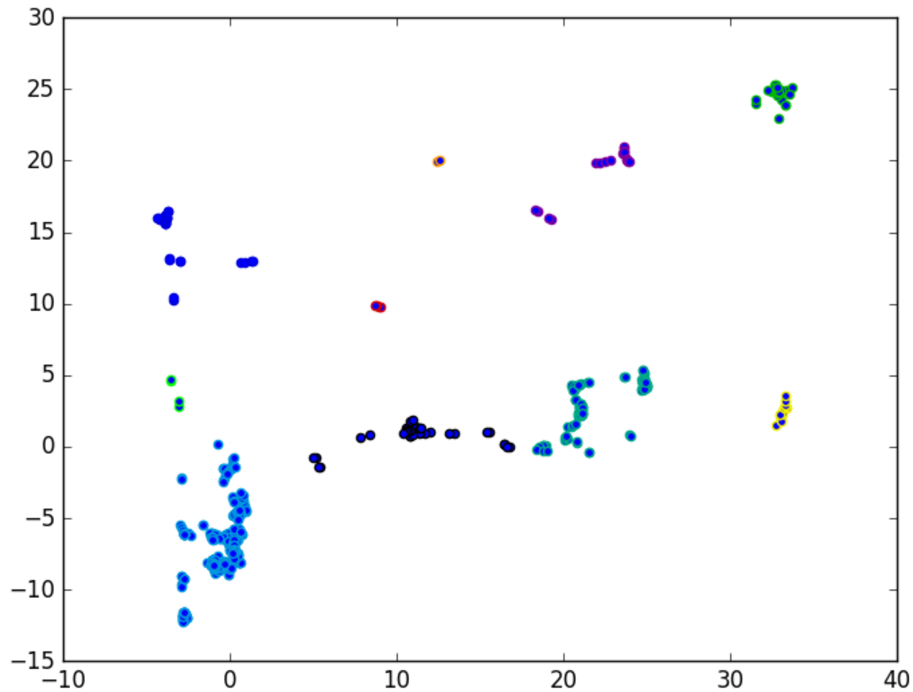

Figure 4.5: Testing our data with Hierarchical Clustering algorithm

with different data the number of clusters may be varies. Comparing Figure 4.1 and Figure 4.5 we can find that the result is not bad. Basically, the office, the restroom and the entrance are detected. However, there are some other clusters which are locations where user always visit. This is because common Connectivity based clustering algorithm can't get rid of noise, in anther word, every point in the data set should be put in to one cluster.

4.1.2 Centroid-based clustering algorithms (k-means clustering)

How does k-means cluster algorithm work?

As the name says, there will be k centroids among data points. In addition, each centroid will be surrounded by one cluster and the centroid is the centroid point of that cluster. So before starting, the value of k should be given as input. At first, this algorithm places K centroids $c_1 \ldots c_k$ at random position among data. Then this algorithm keeps repeating the following two steps: 1. Assigning points to cluster whose centroid is closest to the point. 2. Calculating the main position of each cluster and let the main position be the new centroid point of each cluster.

Pseudo-code 4.6 shows the basic structure of this algorithm.

```
K-MEANS (Data)
while(Cluster[] is changed)
    (1)foreach x_i in Data
            find the nearest centroid c_j
            assign x_i to Cluster[j]
    (2)foreach Cluster[j] in Cluster[]
            c_j = mean(x ∈ Cluster[j])
```

Figure 4.6: Pseudocode for K-means clustering algorihtm.

What kind of data can be handled by K-means clustering algorithm?

Because the value of k should be given at first. So the amount of clusters which data will be clustered into should be known. What's more, data should surround each centroid point. In our case, if k is known the k-means clustering algorithm seems to be a good solution.

Testing K-means clustering algorithm with user's location data:

Figuer 4.7 shows the result of testing the K-means clustering algorithm with location data gathered from the wheelchair. We select the value of k to be 10. Comparing the result with the map in Figure 4.1, we find the result is not as good as expected. Those points colored red and dark blue are connected. And the density of them is relatively higher then the other regions. However, they are divided into

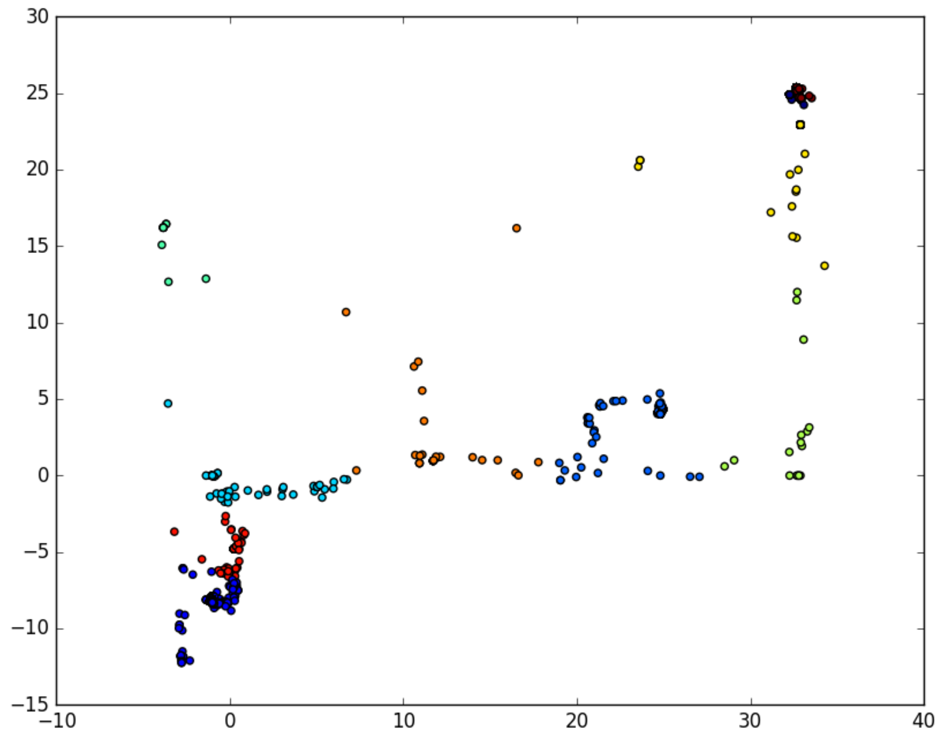

Figure 4.7: Testing result of K-means Clustering algorithm

two clusters. For those light yellow points, the are not connected and divided by a region which has no point inside. But they are still in one cluster. The reason of this phenomenon is that 10 is not the perfect value of K. A little change in K's value can lead to a totally different result. What's more, K-means Clustering algorithm can't find the global optimal solution. The result of this algorithm is also depends on the position where k centroid points are put at first. Those evidence shows that this clustering algorithm should not be used in our situation.

4.1.3 Distribution-Based Clustering Algorithms (Gaussian Mixture Model)

How does Gaussian Mixture Model work?

The expectation-maximization (EM) algorithm is usually used to get Gaussian distributions from the given data set. The basic idea of EM is to pretend that we know the parameters of the model and then to infer the probability that each data point belongs to each component. After that, we refit the components to the data, where each component is fitted to the entire data set with each point weighted by the probability that it belongs to that component. The process iterates until convergence[9].

For Gaussian mixture model, random parameters are given to all those Gaussian functions we need to fit before the algorithm starting. The E step is to compute the probability of each point was generated by each cluster. And in the M step we calculate the new σ(the standard deviation of the Gaussian function), μ(the average of the Gaussian function) and w(the weight of specific Gaussian cluster) for each Gaussian function.

What kind of data can be handled by Gaussian Mixture Model?

Any kind of distribution can be modeled by the mixture of Gaussian distributions. However, too many Gaussian distributions many make the model useless. So the model data should be mixture of Gaussian distributions. What's more, the number of clusters is required too. So if we want to use Gaussian Mixture Model on the data gather from wheelchair, we have to develop an algorithm to calculate the cluster number first.

Testing Gaussian mixture model clustering algorithm with user's location data:

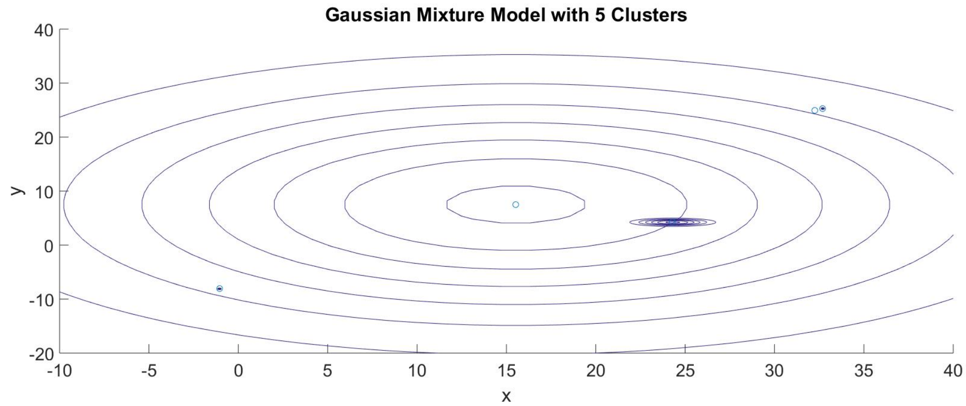

Figure 4.8: GMM with 5 clusters

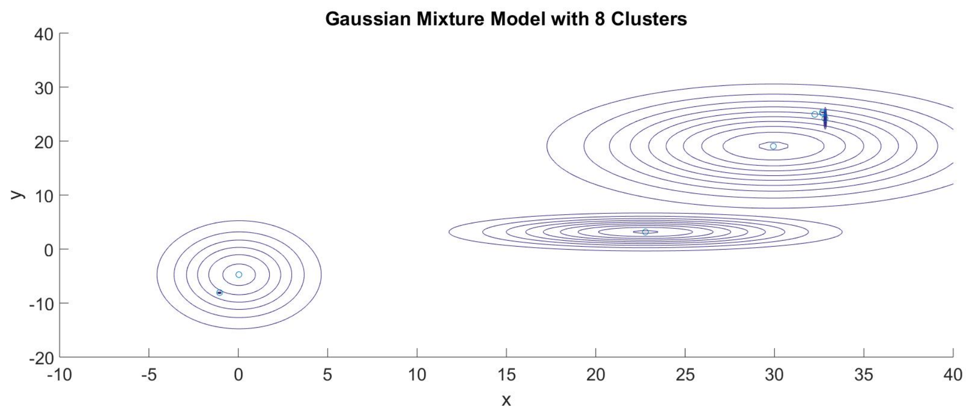

Figure 4.9: GMM with 8 clusters

Figure 4.8 to Figure 4.11 shows the GMM result with different numbers of gaussian clusters. In the result we find, with small cluster numbers, Gaussian Mixture Model can't find all clusters we need. What's more, the Gaussian model find two clusters at the entrance place where we expect only one cluster. With larger cluster number, this algorithm can find all those clusters we need. However, many clusters overlap together, which means in the same location this algorithm get multiple clusters. We have to develop another algorithm to combine all those cluster together.

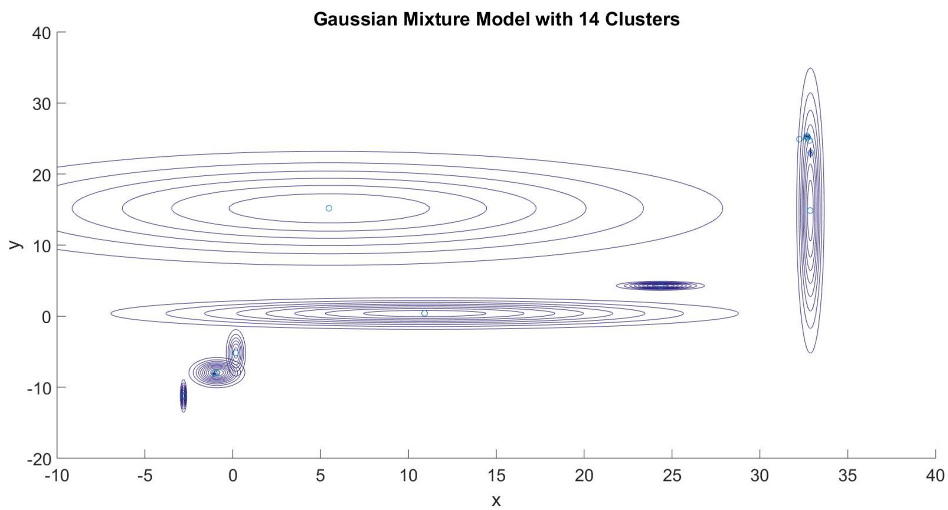

Figure 4.10: GMM with 14 clusters

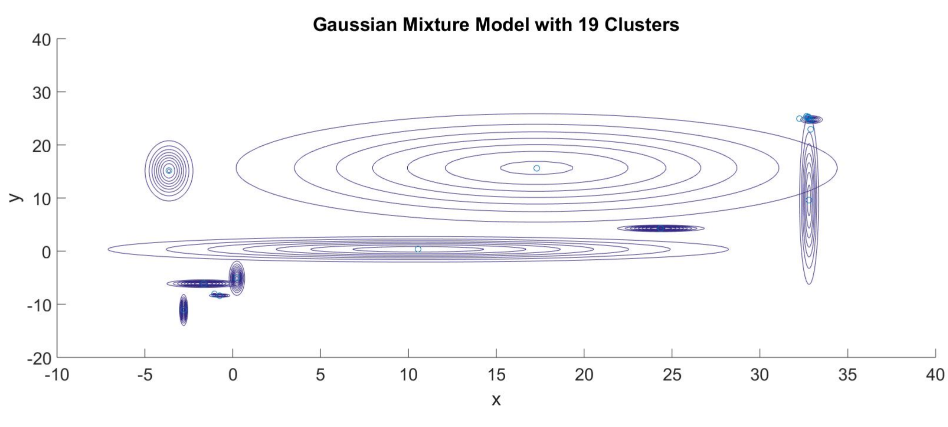

Figure 4.11: GMM with 19 clusters

4.1.4 Density-based Clustering Algorithms

How does Density-based clustering work?

As the name density-based clustering, this algorithm requires the density as input. There are two parameters to determine the density. One is , the radius of the circle in which the density will be calculated. Another one is the minimum number of points (minPts) required to be inside the circle.

$$Density = \frac{minPts}{2\pi\varepsilon^2}$$

The basic idea of this algorithm is to make points which are adjacent and the density of them is larger than given density a cluster. For those points which have less points around will be treat as noise.

What kind of data can be handled by Density-Based Clustering algorithm?

The only requirement for this algorithm is that the lowest density of each cluster have to be known. It is also the best feature of Density-Based clustering algorithm. By contrast, all the other clustering algorithms mentioned above needs the amount of clusters as input. With this feature, Density-Based clustering algorithm can find the amount of clusters for the other clustering algorithm.

Testing Density-Based clustering algorithm with user's location data:

Figure 4.12 shows the result of using Density-based clustering algorithm on the location data of the wheelchair. In this figure we can find that lots of point are disappeared. It is because that the densities of the points surrounds those points are lower than the threshold. So those points are treated as noise. It make sense. Locations where the wheelchair visit occasionally should not be considered as locations of interest. In addition, locations like kitchen, office, restroom and entrance are all found.

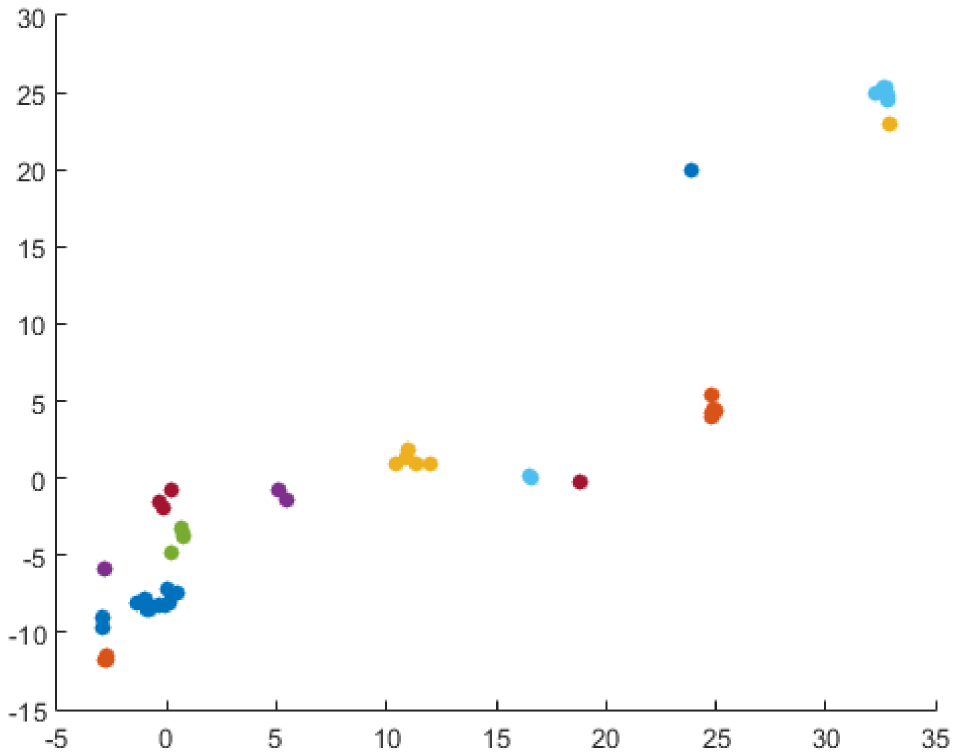

Figure 4.12: DBSCAN with location data of wheelchair

4.2 Clustering Algorithm Used in This Research

In this project, we need to divide those location points into clusters as well as generate the probability distribution of those location points. None of the algorithm mentioned above can achieve this goal. So we combined the Revised Density-Based Spatial Clustering of Applications with Noise (revised DBSCAN) algorithm from [13] and Gaussian Mixture Model (GMM) to be the solution for this project. The DBSCAN is applied at first to divide the location points into clusters. After that, the algorithm will calculate the time how long have the wheelchair stayed in each cluster. Clusters with staying time less than the threshold will be discarded. Then inside each remaining clusters, GMM is applied to get the probability distributions of each clusters got from last step. Note that the GMM is not used to divide location points into clusters. It only used to get the distribution of points inside each cluster which we get from the DBSCAN step. Figure 4.13 shows the result of the combination of this two algorithm. Those colored ellipses are clusters with probability distribution. Different color means different probability density.

Compare Figure 4.13 and Figure 4.1 we can find that this solution works pretty well. Kitchen, office, restroom and entrance are all found. The other two clusters, one is in the front of office another one is at the corner of the corridor, seem to be errors because they don't belong to any room. Actually, they are also useful locations. During our experiment we find that the wheelchair have to stop at the front of the office to wait for the other person to open the door. What's more, because of the size of the wheelchair, it have to stop at the corner of the corridor to adjust its direction to pass the corner. So wheelchair have to stop at those two locations in daily life. That makes those two locations important destinations too.

In next chapter, the method used to learn the relation between locations and time as well as a short introduction of Bayesian Network is given.

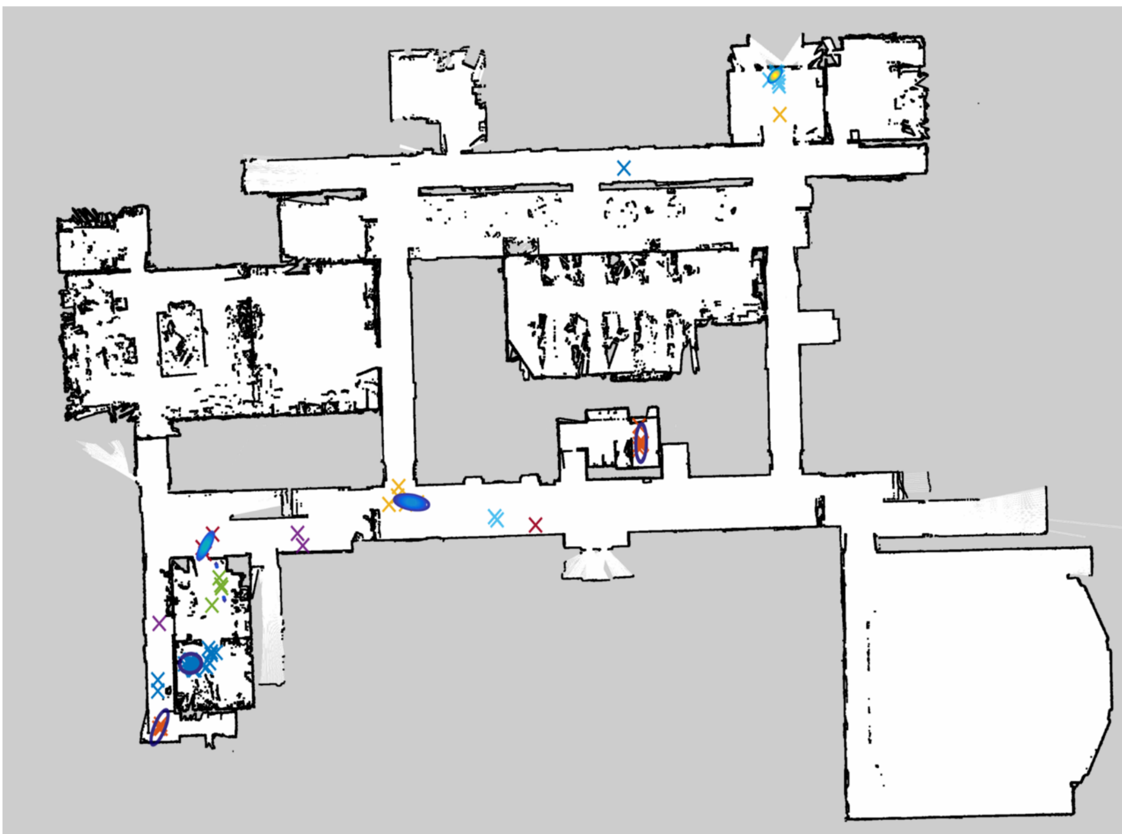

Figure 4.13: Final result of clustering

Chapter 5

Learning Relationship between Locations Using Bayesian Network

Regions where the user usually visit is gathered in last chapter. In this chapter, I'll discuss how to find relationships between locations and predict the user's future location based on the history data of when the wheelchair visited each region. Several methods are tried, such as calculating the covariance between each time when each region was visited, performing Fourier transform on time-presence data of each region to get main frequencies and learning Bayesian network based on the data of when each region is visited. After trying all those methods, we select Bayesian network as our main method to fulfill this task of .

In the following sections, I will provide a short description of how Bayesian network is selected at first. Then follows a brief introduction of Bayesian network in section 5.2. After that, details of learn relationship between locations by using discrete Bayesian network will be provided.

5.1 Method Selection

"Predicting human's behavior" sounds like impossible. One person can change his or her mind so quickly. And it is still a question whether human's behavior can be modeled by any kind of simple statistical model. What's more, we find very little research on this field. So we tried lots of method to solve this problem. Most of those methods cannot work. However, it is helpful to introduce those filed methods. Because, those methods gave us lots of clues for this problem. In the following paragraphs I'll describe methods we have tried in time order.

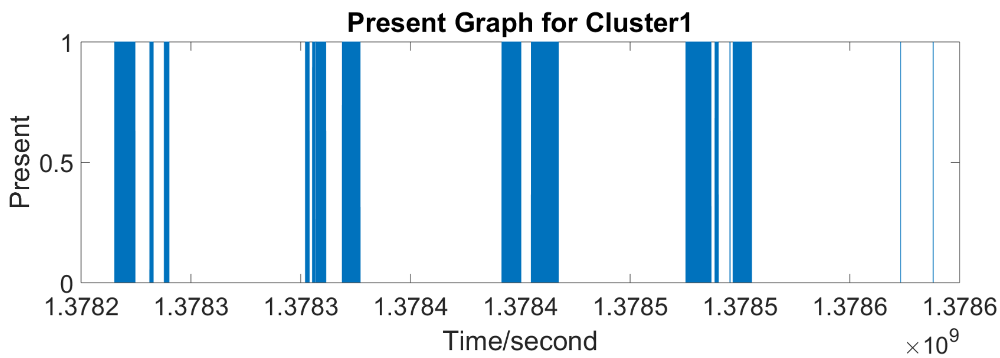

Figure 5.1: Presenting Graph for Cluster1 (office)

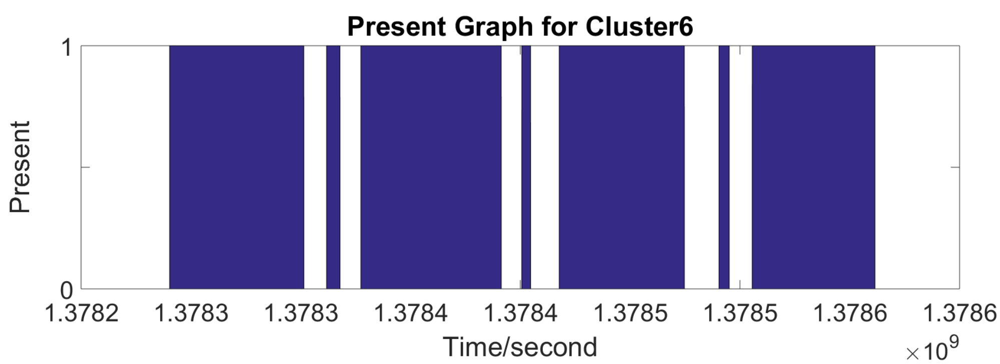

Figure 5.2: Presenting Graph for Cluster6 (entrance)

If we track all those activities of a person for a long time. It is not hard to find that most of those activities repeat day by day or week by week. It gives us a clue that human's activities may be periodical. The first idea came to our mind is to detect the period for each kind of event. Then human's behavior can be model using some main periods. To test this idea, we generated 0 and 1 data which indicate whether the wheelchair is in the specific region from last chapter. 0 means the wheelchair is not in the specific region and 1 means the wheelchair is there. This kind of data is called "**Presenting Data**". Figure 5.1 and Figure 5.2 in which 1 points are colored blue and 0 points are colored white can give us a better view. Cluster 1 is the office, and cluster 6 is the entrance.

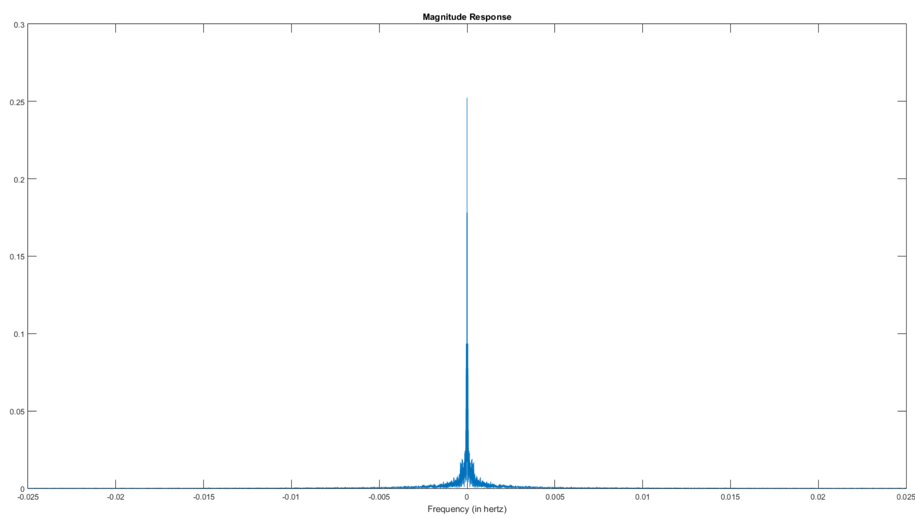

Figure 5.3: FFT of Present Data

Selecting cluster 1 (office) as example, to find the period, we performed a Fourier transform on the presenting data. The result is shown in Figure 5.3 to Figure 5.5. The first graph is the result of the Fourier transform. And the other three graphs are presenting to show the coordinate of two main peaks.

From Figure 5.3 to Figure 5.5, we can find that the frequency of the second highest peak is $1.073 \times 10^{-5} Hz$, the cycle length is 25.88 hours. We can ignore

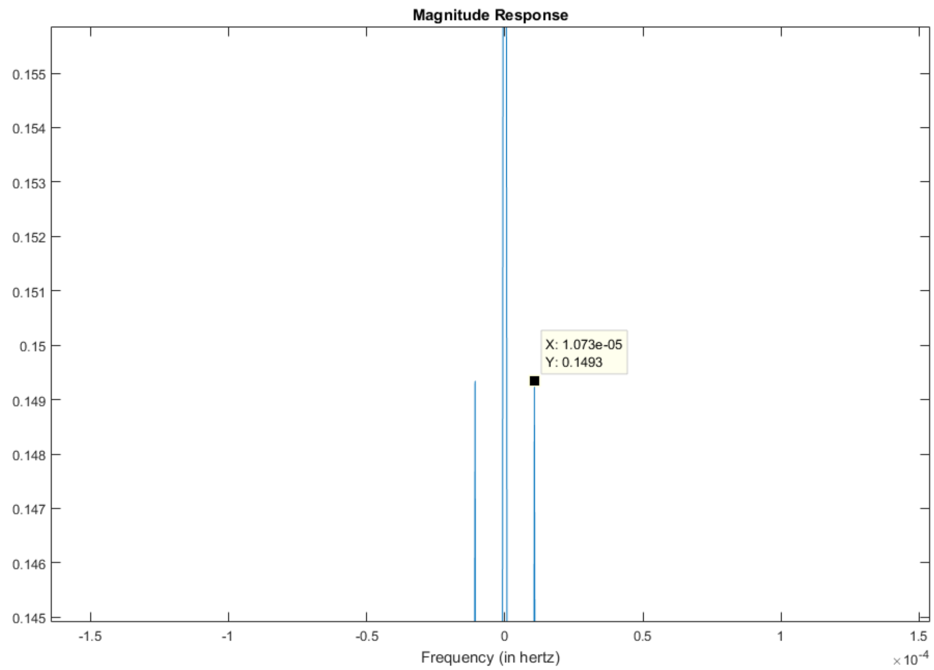

Figure 5.4: Second Peak of FFT

the first peak whose frequency is 0. For cluster 5 (entrance) the cycle length is 22.31 hours. For all locations the second highest peak's frequency value are around $1.16 \times 10^{-5} Hz$, the cycle length of which is 24 hours. This gave us a very useful clue that the main cycle length for almost every human's activity is one day.

Using the value of these main peaks, we can perform the inversed Fourier transform. The result is shown in Figure 5.6.

Comparing this result with the presenting graph of cluster 1 in Figure 5.1, we can find that the result is not as good as we expected. The result not only suffers from inaccuracy but also has some over-fitting problem. Using the result, the user will keep repeating the history. There are several reasons for this failure: 1. This model does not consider the relationship between different nodes. 2. The activity within one day is not totally periodically. 3. Sleeping acts like a reset button which

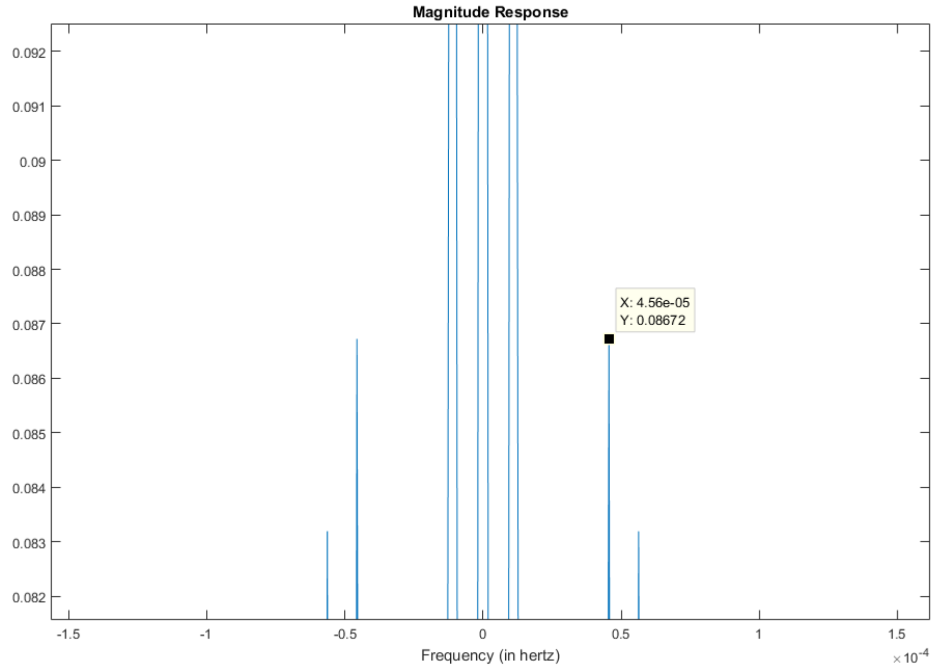

Figure 5.5: Third Peak of FFT

interrupt those periodical activities.

Even though this method is not usable, we still learned something helpful. Since the main period of all those events is one day, we can average data from a long time into one day. Figure 5.7 shows the probability of the wheelchair being in cluster1, which is the office of the PhD student who attend the experiment. This result shows that the probability of the wheelchair being in the office is quite related to the time of the day.

However, for some locations like restroom where seems to be visited randomly, this time of the day method will not work. To deal with that, we come up with an idea, to find the relationship between locations. So we studied the transform between locations and made a Markov chain in Figure 5.8 based on those transforms. This Markov chain shows the probability of the wheelchair going from one location to

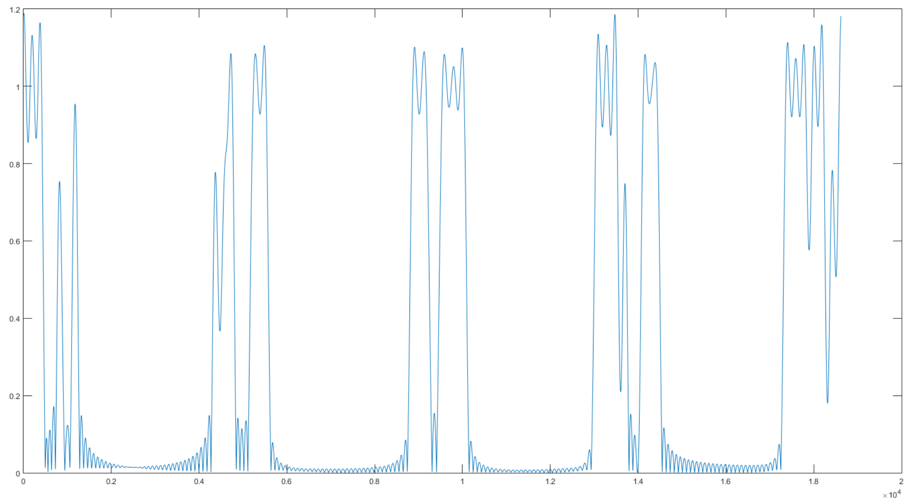

Figure 5.6: Result of Inversed Fourier Transform

another location. Each letter in this graph represents one location. A is the office. B is the restroom. C is the small kitchen near the office. D is the corner of the corridor and E is the entrance of the building. The arrows point to the destination where the wheelchair goes to. The number on each arrow is the probability of the wheelchair moving to the pointed location. We can find that every time the user entered the building, the place he went to is the office. What's more, after the user visited the kitchen, he headed back to office directly. This result shows that there is strong relationship between locations.

So far, we can make the conclusion that the probability distribution of where the wheelchair heads to may depends on two factors. One is the time of the day. The other one is the locations where the wheelchair have visited that day. To combining these two factor, we selected Bayesian Network as the basic method. Because Bayesian Network can study the probabilistic relations between different variables.

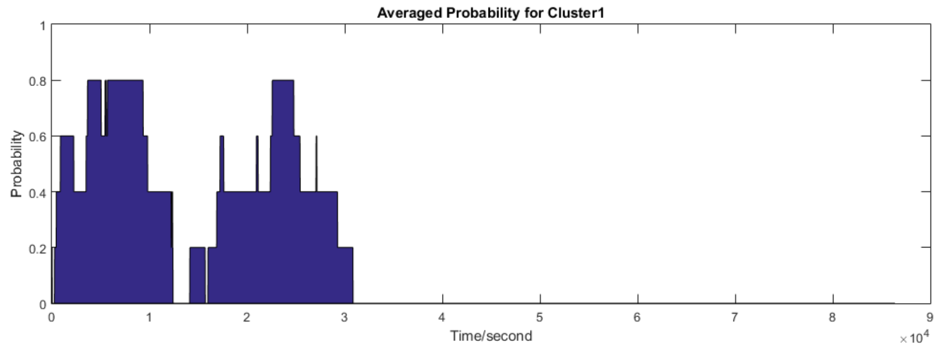

Figure 5.7: Probability of being in Cluster 1 among one day

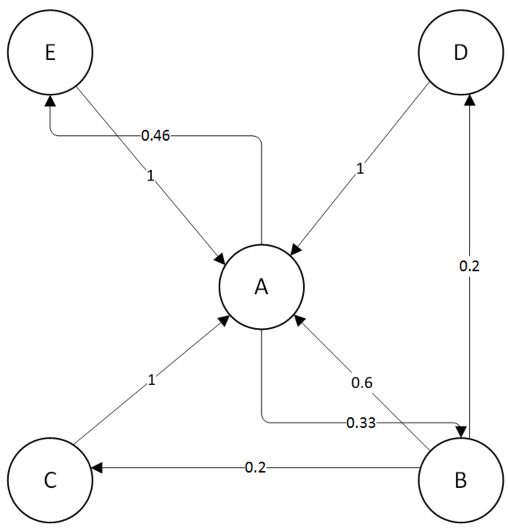

Figure 5.8: Markow chain between each location

5.2 Introduction of Bayesian Network

In the following paragraphs, I present a short introduction of Bayesian Network. At first I show some necessary mathematic backgrounds. Then follows the basic structure of Baysian network and how to use the Bayesian network. Thirdly, I show methods for learning parameters of Bayesian network. And finally, I give a short introduction of methods used for learning Bayesian network's structure.

5.2.1 Preliminaries: Math Background

What is probability? The intuitive understanding of probability is how likely the given event will happen in a given situation or the long run average of a repeating random experiment. However, to study and use probability more easily, a more rigorous definition should be given.

At first, the concept of probability space will be introduced. A probability space, which is a triple (Ω, F, P), consisting of the following three elements:

1. A sample space Ω, which is composed by all of the possible outcome of the random experiments.

2. An event class F. F is a nonempty collection of the subsets of Ω.

3. A probability measure P which maps F to real number.

Since each event is a subset of Ω, the probability of two events A and B happening together should be $P\left(A \cap B\right)$. And the conditional probability of the event B occurring given the condition that A is happening is given by:

$$P\left(A \mid B\right) = \frac{P(A \cap B)}{P\left(B\right)}$$

This equation is identity to

$$P\left(A \cap B\right) = P\left(A \mid B\right) * P\left(B\right) = P\left(B \mid A\right) * P(A)$$

The equation above implies:

$$P\left(B \mid A\right) = \frac{P\left(A \mid B\right) * P(B)}{P(A)}$$

This equation is the *Bayes' formula*. *We* can also get the *chain rule* from those equations above:

$$P(A \cap B \cap C) = P(A) * P(B \mid A) * P(C|A \cap B)$$

By using the *Bayes' formula* and the chain *rule* we can get the following equation easily.

$$P(A \mid B \cap C) = \frac{P(A) * P(B \mid A) * P(C|A \cap B)}{P(B \cap C)}$$

In the equation above,

$$P(B \cap C) = P(A \cap B \cap C) + P(\overline{A} \cap B \cap C)$$

Those equations mentioned above are very important in the Bayesian Network.

5.2.2 Bayesian Network

A Bayesian network is a directed acyclic graph (DAG) which represents a probability distribution over a set of variables. It consists of following two parts:

1. The directed network structure in the form of directed acyclic graph.

2. A set of the local probability distributions, one for each node/variable, conditional on each value combination of the parents. [4]

An example of Bayesian network is shown in the Figure 5.9.

With Bayesian network we can inference the probability of any node or any combination of any node. For example, in the Bayesian network shows in Figure 5.9:

$$P(ABC) = P(A) * P(B) * P(C|AB)$$

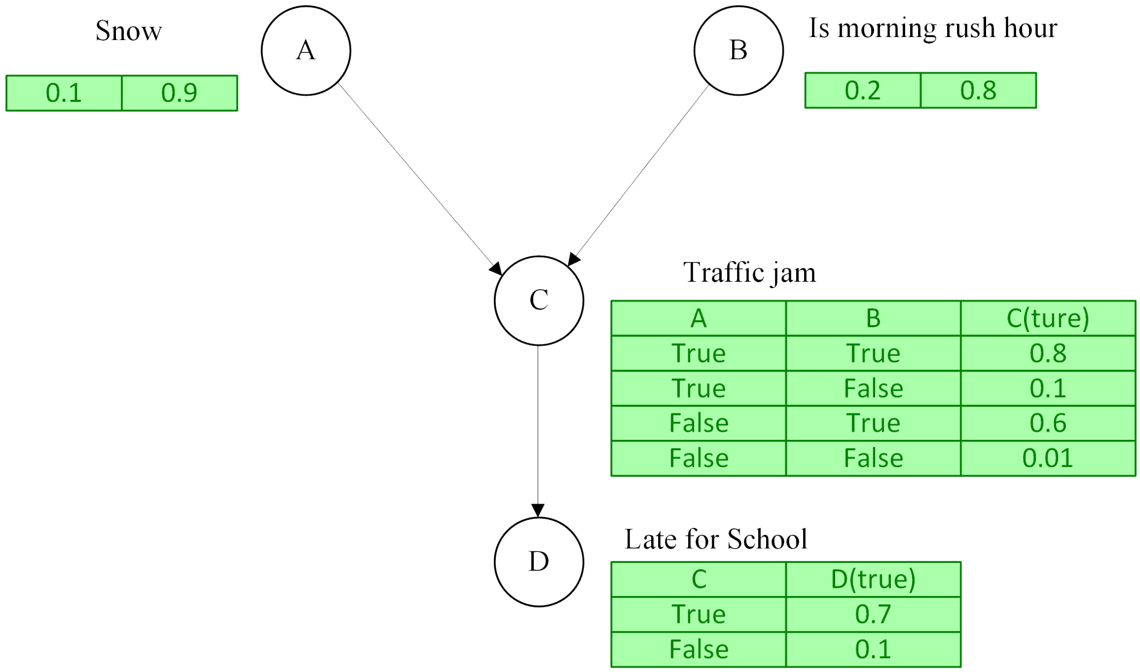

Figure 5.9: Markow chain between each location

$$P(C) = P(ABC) + P\left(\overline{A}BC\right) + P\left(A\overline{B}C\right) + P\left(\overline{A}\overline{B}C\right)$$

$$= P(A) * P(B) * P(C \mid AB) + P\left(\overline{A}\right) * P(B) * P\left(C \,\middle|\, \overline{A}B\right) +$$

$$P(A) * P\left(\overline{B}\right) * P\left(C \,\middle|\, A\overline{B}\right) + P\left(\overline{A}\right) * P\left(\overline{B}\right) * P(C|\overline{A}\overline{B})$$

From the example shown above, we can conclude that for probability like $P(ABC)$ can be get directly from the Bayesian network by multiply the value in each node's probability table. I call this kind of probability "direct probability" And for probability like $P(C)$ which we can't get by multiplying the probability of each node can be get by summing up direct probabilities to get a marginal probability.

5.2.3 Learn Parameters of Bayesian Network

For most problems, we don't have the Bayesian Network ready to use. So learning Bayesian Network form data becomes a very important task even though it is a really challenging one. As I mentioned before, one Bayesian network is consisted with two parts, they are a directed network structure and the local probability distribution for each node. To learn the whole Bayesian Network, researchers usually start with solving the problem of learning the local probability distribution for a fixed structure. This is because many structure learning algorithms estimate parameters as part of their processes.[2]

For each node inside Bayesian network, the processes of learning the local probability distribution function for each node are totally identical. So in the following paragraphs I'll focus on learning local pdf for one node.

At first, let's look at the method of learning parameters for discrete Bayesian Network, all nodes of which have finite outcomes. This task is mainly to fit parameters to a model. To solve this kind of problem, Bayesian approaches seem to be dominant. The main idea of Bayesian approaches is giving a prior pdf first, then use the data gathered through experiment to get the posterior pdf based on the Bayes' rule shown below. And use the posterior pdf to update the prior pdf.

$$P(H = h \mid E) = \frac{P(E \mid H = h) * P(H = h)}{P(E)}$$

Where "H" stands for any hypothesis we want to study and whose probability may be affected by data. And "E" stands for the evidence from which we will learn these parameters.

$P(H = h)$ is the prior probability which means the probability of H = h is true before E is observed.

$P(E \mid H = h)$ is the likelihood of the data given the condition that $H = h$.

$P(E)$ is also called marginal likelihood. This factor is the marginal probability of E by summing up all those conditional probability of E under all possible H.

$P(H = h \mid E)$ is the posterior probability of $H = h$, which we what to get from the data.

To update prior pdf with the posterior pdf the form of those two pdf should be the same. This kind of distribution family is called conjugate prior to a data distribution when the posterior over the parameters belongs the same family as the prior, albeit with different hyper-parameters.[4]

For discrete Bayesian network, we can assume that the outcome of each node is a random variable with multinomial distribution, which can take on one of a finite number of possible values. Learning the parameter of multinomial distribution, we should use Dirichlet family as the conjugate prior. If the size of the data set is large enough, we can just use the normalized frequency as the probability. Having the conjugate prior, the following step is to update the probability distribution function with data to get posterior probability distribution. In our case the probability distribution of each node is binomial distribution, so the Dirichlet family becomes Beta family.

A lot of literature on Bayesian networks assumes data of each node are multinomial. However, for many application, the data supplied are continuous and therefore ways must be found to handle this situation. The easiest way to deal with continuous data might be to discretize the data. In this thesis's section 5.3, I used this technique to divide time data in to sections by fixed time period. For example, treat the time point inside the same hour as the same time. This method has several disadvantages, such as, the accuracy is low for some kind of data and it may need huge computational work. The other methods for representing continuous data are usually give an assumption of how data distribute at first. The most common assumption is that data are under normal distribution. So the problem of learning parameter of one node transformed to learn the parameter of the normal distribution. There are lots of researches on this field. Expectation-Maximization algorithm will solve this problem easily. However, not all kinds of data are normally distributed. Finding the right statistical model for the data becomes the most important task. In section 5.3 of this thesis, I'll use Gaussian Mixture Model as the statistical model for the data. John and Langley show that the non-parametric approach is also an option [2].

5.2.4 Learn the Structure of Bayesian Network

There are mainly two kind of methods to learn the structure of Bayesian network. One is score-based method; another is constrain-based method. Score-based method basically set a score function at first and then calculate the score for given Bayesian network. Based on the score, algorithms such as hill climbing or simulated annealing will be applied to change the structure of the Bayesian network. And finally, we can find one Bayesian network which has the lowest (in some case may be highest) score. Then return that Bayesian network as result. In constrain-based methods, at first, we need to get the relationship between each node by doing conditional independence test. However, only the independence testing is not enough, several assumptions need to be made. Those assumptions are Faithfulness, Causal Sufficiency and Causal Markov. With those constrains the final structure of Bayesian network can be generated. In this thesis the score based method is chosen. And the score function is selected as Bayesian Information Criterion (BIC). The detail of this learning process will be introduced in section 5.3 and section 5.4.

5.3 Discrete Bayesian Network for Learning Human's Habit

As discussed in the method selection section, the probability for the wheelchair of being at one specific location depends on both the time of the day, what locations are visited before and when the wheelchair left those locations. And Bayesian network may be a good way to represent the relationship between them. However, most algorithms for learning Bayesian network from data are designed for discrete data. So we discretize those continuous data at first, then applied algorithm mentioned before to learn Bayesian network.

The main process of learning relationship between locations using discrete Bayesian network is shown in the Figure 5.10. In this section I'll introduce this whole system by the order of this flow chart. The first step is to generate the data for learning. An example of data used for learning will be provide. With the data, the second step is to learn the parameter for the Bayesian network with given structure. It is worth to mention the method used to deal with the values out of the boundary. Having the parameter and the structure of the Bayesian network, we can calculate the Bayesian Information Criteria (BIC) which I discussed in last section. Then the only thing we need to do is keep changing the structure of the Bayesian network until the minimum BIC is found. To speed up this process, algorithms like hill climbing or simulated annealing can be used to change the Bayesian network's structure. Then return the Bayesian network who has the minimum BIC as the final result.

5.3.1 Data Preparing

Table 5.1 shows an example of data used in this learning process.

The data is gathered from a PhD student who use this wheel chair every day for

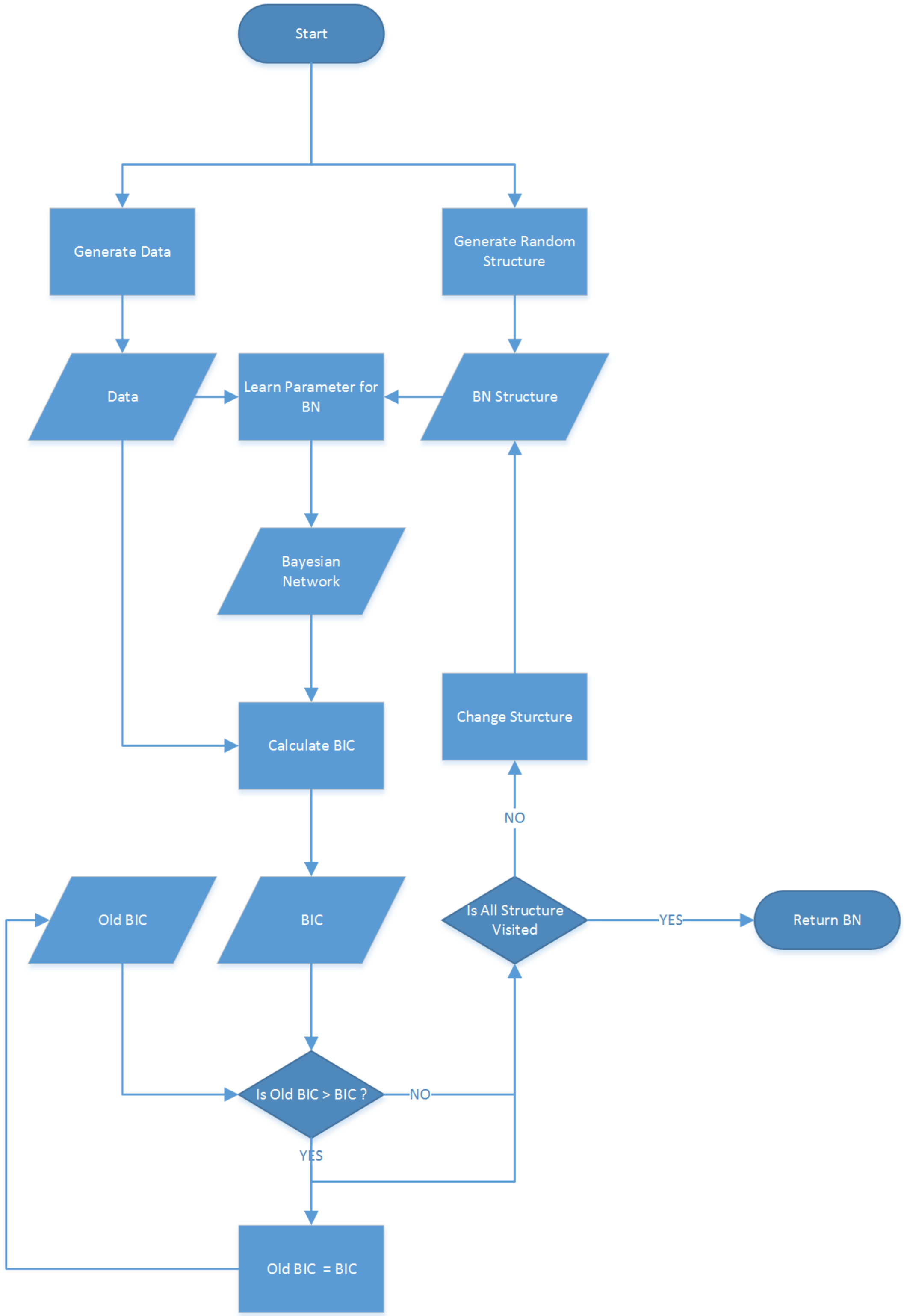

Figure 5.10: Flow Chart of Discrete Bayesian Method

T	Ta	Tb	Tc	Td	Te	A	B	C	D	E
41955.47	4411.868	6652.96	80471.96	73687.46	9763.323	1	0	0	0	0
41975.47	4431.868	6672.96	80491.96	73707.46	9783.323	1	0	0	0	0
41995.47	4451.868	6692.96	80511.96	73727.46	9803.323	1	0	0	0	0
42015.47	4471.868	6712.96	80531.96	73747.46	9823.323	1	0	0	0	0
42035.47	4491.868	6732.96	80551.96	73767.46	9843.323	1	0	0	0	0
42055.47	4511.868	6752.96	80571.96	73787.46	9863.323	1	0	0	0	0
42075.47	4531.868	6772.96	80591.96	73807.46	9883.323	1	0	0	0	0
42095.47	4551.868	6792.96	80611.96	73827.46	9903.323	1	0	0	0	0
42115.47	4571.868	6812.96	80631.96	73847.46	9923.323	1	0	0	0	0
42135.47	4591.868	6832.96	80651.96	73867.46	9943.323	1	0	0	0	0
42155.47	4611.868	6852.96	80671.96	73887.46	9963.323	1	0	0	0	0
42175.47	4631.868	6872.96	80691.96	73907.46	9983.323	1	0	0	0	0
42195.47	4651.868	6892.96	80711.96	73927.46	10003.32	1	0	0	0	0
42215.47	4671.868	6912.96	80731.96	73947.46	10023.32	1	0	0	0	0
42235.47	4691.868	6932.96	80751.96	73967.46	10043.32	1	0	0	0	0
42537.15	301.6751	7234.635	81053.64	74269.14	10345	0	0	0	0	1
42557.15	321.6751	7254.635	81073.64	74289.14	10365	0	0	0	0	1
42577.15	341.6751	7274.635	81093.64	74309.14	10385	0	0	0	0	1
42597.15	361.6751	7294.635	81113.64	74329.14	10405	0	0	0	0	1
42617.15	381.6751	7314.635	81133.64	74349.14	10425	0	0	0	0	1
42637.15	401.6751	7334.635	81153.64	74369.14	10445	0	0	0	0	1
42657.15	421.6751	7354.635	81173.64	74389.14	10465	0	0	0	0	1
42677.15	441.6751	7374.635	81193.64	74409.14	10485	0	0	0	0	1
42697.15	461.6751	7394.635	81213.64	74429.14	10505	0	0	0	0	1
42717.15	481.6751	7414.635	81233.64	74449.14	10525	0	0	0	0	1
42737.15	501.6751	7434.635	81253.64	74469.14	10545	0	0	0	0	1
42757.15	521.6751	7454.635	81273.64	74489.14	10565	0	0	0	0	1
42777.15	541.6751	7474.635	81293.64	74509.14	10585	0	0	0	0	1
42797.15	561.6751	7494.635	81313.64	74529.14	10605	0	0	0	0	1
42817.15	581.6751	7514.635	81333.64	74549.14	10625	0	0	0	0	1

Table 5.1: Example Data for Learning Discrete Bayesian Network

experiment. I only chose 5 locations to explain this whole idea briefly. Location A is the laboratory where the PhD works in. Location B is the restroom. Location C is the small kitchen where coffee can be cooked close to the laboratory. Location D is one meeting room. And Location E is the entrance of the ECE department building.

The first column in the table above named as T which means the time of the day. The range of T's value is between 0 to 24 hours (86400 seconds). Values in column from Ta to Te indicates how long ago the wheelchair had left the specific

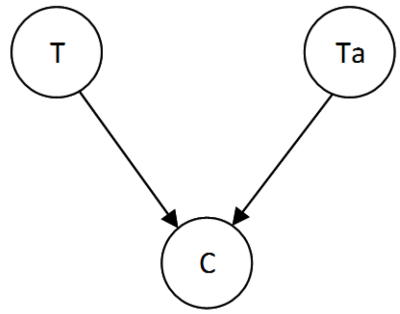

Figure 5.11: Flow Chart of Discrete Bayesian Method

location. And values in column A to E indicates whether the wheelchair is in the specific location under the condition given by the first six columns. 1 indicates the wheelchair is there, and 0 means wheelchair is not there. And this data was gathered from the wheelchair every 20 seconds. To show ideas clearly and conveniently, each column from T to Te will be called one **dimension**.

5.3.2 Parameter Learning

This subsection mainly discusses about the method used to learn the probability mass function for each single node. The main task for parameter learning is to generate a table of conditional probabilities. To demonstrate the method for parameter learning, one example will be shown in the following paragraphs. Figure 5.11 shows the structure of the node. C is the node that we want to study.

An example of the conditional probability table we what to learn is shown in Table 5.2. The value in each cell is the conditional probability of given condition. For example, the value cell marked red is equal to $P(C|T=3,\ Ta=2)$, in the other words, the conditional probability when time of the day is 3 o'clock and the it's 2 hours since the wheelchair left location A. There are two kinds of nodes can be the parents of any node. One is the time of the day, and anther is how long ago the

	1	2	3	4	...	24
1	0.4	0.32	0.2	0.1	...	0.01
2	0.3	0.2	0.71	0.22	...	0.03
...	...	...	...	...	...	...
6	0.1	0.08	0.01	0.01	...	0.01

Table 5.2: example of the conditional probability table

wheelchair has left the specific location. The range of the former one is between 0 to 24 hours. And the range of the latter is 0 to infinity. However, for the practical aim, the boundary for the latter one should be a finite number other than infinity. So we assume that there is no strong relationship between events the interval between whom is larger than 6 hours. With this assumption, we can set the boundary of the how long ago the wheelchair has left the specific location to be from 0 to 6 hours.

Because of the size of the data set, Bayesian approach will be used to learn the value in each cell. The main idea of this method is. Assigning a prior probability for each cell. Then update this probability by using data piece by piece. The prior we selected is the marginal frequency for each node. By marginal frequency, I mean divide the whole number of 1s in the column of specific location by the whole number of data entry. And then the posterior probability is

$$\frac{Prior \ \bullet Prior's \ weight + count \ of \ 1s \ for \ given \ condition}{Prior's \ weight + count \ of \ 0s \ for \ given \ conditions}$$

For example, let's learn the probability of C = 1 under the condition of T = 3 and Ta = 2 from data shown in Table 5.3 The prior probability for the node C is 4/7. And $\mathrm{P(C = 1|T = 3, Ta = 2)}$ (the conditional probability for that cell) = $(4/7 * prior'sweight + 3)/(prior'sweight + 4)$. If the prior's weight is 1, then the posterior probability for that cell is 0.71.

However, the value of some dimension of some data will be larger than the

T	Ta	C
3	2	1
1	3	0
3	2	0
2	3	1
3	2	1
2	1	0
3	2	1

Table 5.3: example data

upper boundary. We can't just ignore those data. Because that most of this kind of data have only one dimension out of the boundary. And for those data the other dimensions are still useful. For this case, one data point will be divided into several points uniformly distributed on dimensions which have data out of boundary. Assuming weight of original each data point is one, then the weight of those divided points are 1 divided by how many pieces the point is divided into. The following equation shows how to calculate probability for each cells in this situation:

$$\frac{Prior \bullet Prior's\ weight + \sum_{points} wight * point's value}{Prior's\ weight + \sum_{points} wight}$$

5.3.3 Structure learning using BIC

BIC is the abbreviation of Bayesian Information Criterion. It is introduced by Schwarz as an asymptotic approximation to a transformation of the Bayesian posterior probability of a candidate model.[6] BIC is used to select the best model that can fit the data without overfitting. The lower BIC value means the better the model. The BIC for a given model is defined as

$$BIC = -2\ln L\left(\widehat{\theta}_k \middle| Data\right) + k \bullet ln(n)$$

In the equation above, $\ln L\left(\widehat{\theta}_k \;\middle|\; Data\right)$ is the log-likelihood of the estimated parameter by given data. k is the number of parameters. n is the size of the data set.

In our system, we generate all those possible structure of Bayesian Network at first. Then we use BIC to select the best structure of the Bayesian network. For any given structure, we can use the method from the last section to learn the parameters of each node. After that, we can calculate the likelihood of this model by given data. Then BIC for each structure can be calculated. Finally, we select the structure with the lowest BIC value as the final Bayesian Network.

Figure 5.12 to Figure 5.19 show the learned conditional probability maps with different structure settings. The conditional probability maps are the parameter of each node. In those Figures, T means time of the day. The unit of x axle and y axle are 30 minuets. Figure 5.12 to Figure 5.14 show the relation between the probability of visiting the office, the time of the day and how long ago the wheelchair visited the entrance. The structure that the probability of visiting office is only related to the time of the day get the lowest BIC score. It meets our common sense. People like the PhD student in our experiment tends to work based on strict schedule. They stay in the office at the same time every day. So the probability of being in the office can be predicted only by the time of the day. What's more, different Bayesian network structure between the probability of visiting the restroom, the time of the day, last time leaving the restroom and last time leaving office is shown in Figure 5.15 to Figure 5.19. We can find that the best factor to predict the probability of visiting the restroom is how long ago have the user left the office. The BIC score of using how long ago the restroom is visit to predict the probability of visiting the restroom is close to the lowest score. In common sense, the probability of visiting the restroom should based on how long ago when the person visit the restroom. But in out case, how long ago have the user left the office becomes the best factor. There are 2 reasons for this result. Firstly, we only have five days' data. Maybe the data is not enough to learn the best result. Secondly, in this experiment, the PhD student will always go to the restroom from the office. Then strong relation between restroom and office was built up.

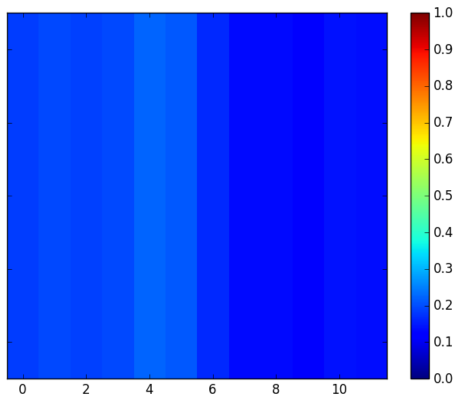

Figure 5.12: Predicting Office using entrance, BIC = 14294

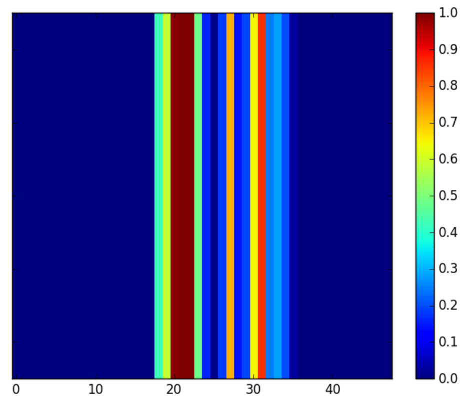

Figure 5.13: Predicting Office using T , BIC = 5928

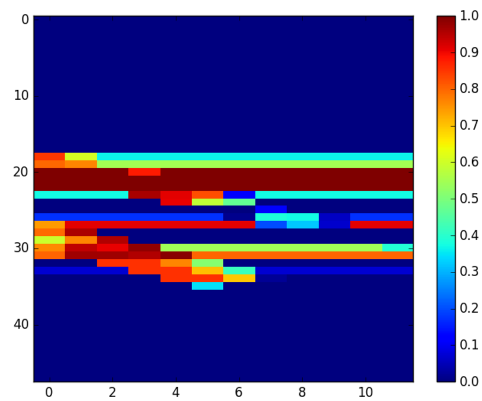

Figure 5.14: Predicting Office using T and entranc, BIC = 10044

5.4 Prediction Using Bayesian Network

Our final goal of this research is to predict the probability of each destination. With the Bayesian Network, predicting becomes easy to achieve. In the following paragraphs I'll describe how the predicting module works.

At first, data used for predicting should be prepared. The data are the time how long ago each location was visited and the current time of the day. Using the time data as input of the Bayesian Network, the probabilities of visiting each destination are calculated. Let's use Figure 5.20 as example. The probability of visiting location C is related to the time of the day, how long ago location A (Ta) is visited and how long ago location B (Tb) is visited. As a example, we can assume T = 10 a.m., $Ta = 1$ hour and $Tb = 3.5$ hours. Because the parameter of this Bayesian Network is learned. How to calculate $P(C|T,\ Ta,\ Tb)$ is known. To calculate the probability of C we only need look up the value of $P(C|T = 10,\ Ta = 1,\ Tb = 3.5)$. However, it is possible that some location haven't been visited by the time when prediction is required. In that situation, assume the location is been visited N and the destination is C, we can use the following equation to calculate the conditional probability of visiting the destination.

$$P(C|T,\ Ta,\ ...) = \frac{\sum_{segments\ of\ Tn} P(C|T,\ Ta,\ Tn,\ ...)}{Amount\ of\ segments\ in\ Tn}$$

Finally, the probability of the destinations are predicted using this algorithm.

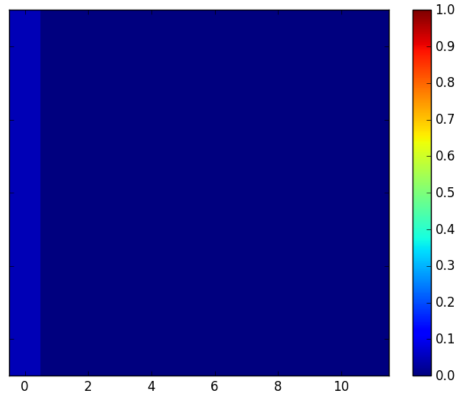

Figure 5.15: Predicting Restroom using office, BIC = 834

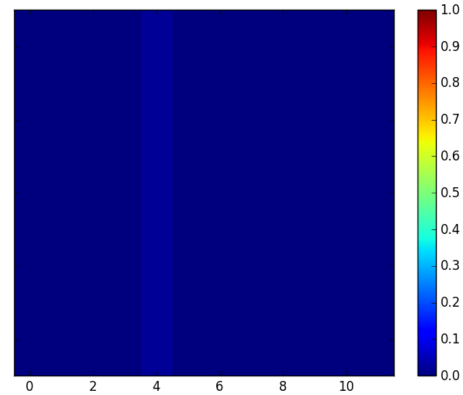

Figure 5.16: Predicting Restroom using restroom, BIC = 1092

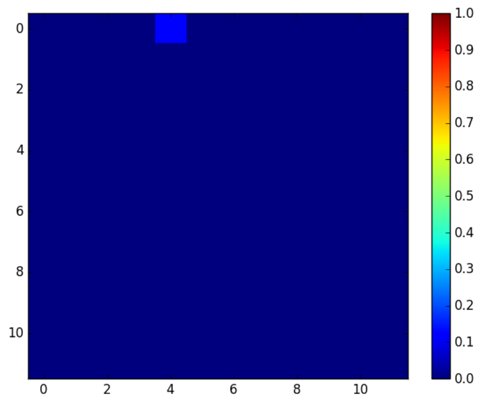

Figure 5.17: Predicting Restroom using Office and Restroom, BIC = 2120

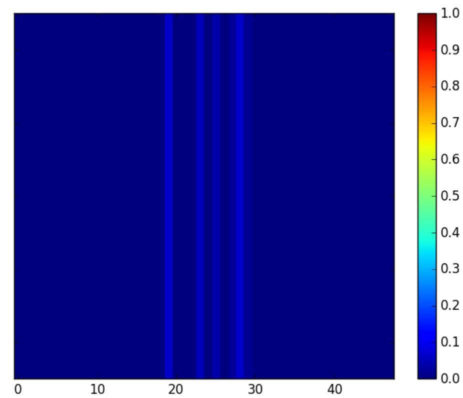

Figure 5.18: Predicting Restroom using T, BIC = 1154

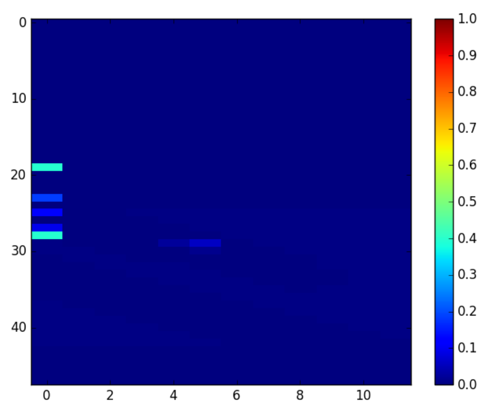

Figure 5.19: Predicting Restroom using the T and Office, BIC = 6010

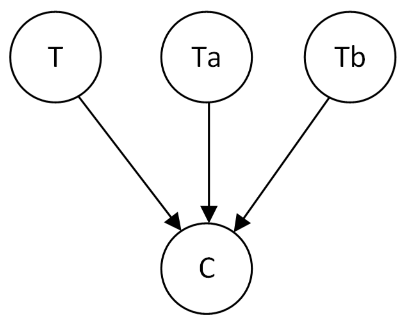

Figure 5.20: Example Bayesian Network

Chapter 6

Conclusion and Future Work

In this chapter I'll conclude the work we have done in this thesis and mention some future work we may do to make this system better.

6.1 Conclusion

In this thesis, I introduced the structure and elements of the system used to learn the relationship between locations where the user have visited respect to time and predict the user's future location. After testing several clustering algorithms on the location data gathered from experiment, combination of DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm and GMM (Gaussian Mixture Model) algorithm is selected to find clusters of locations where user usually visit. Mathematic background and short introduction of Bayesian network is provided in this thesis. With the background knowledge, I explained the Bayesian network used in our system. The result shows that this system have the learning ability we desired.

6.2 Future work

Data collection is almost the hardest part in this whole project. User's location data can only collected from the wheelchair with multiple sensor working together. It means that there have to be some person using this wheelchair everyday. However, we can't find any person with this kind of disability to take part in our experiment. So we have to use the wheelchair by ourselves, which caused us lots of inconvenience. And because of lacking of data, the process of this research is very slow.

As improvement, we should develop a light-weighted localization system which can be used by normal person everyday. This localization system can be an application running on cell phones. And the application can localize itself through the indoor wifi signal and the outdoor GPS signal.

The accuracy and speed of the prediction system discussed in this thesis can still be improved by using continuous Bayesian Network instead of the discrete Bayesian Network. We can assume the probability of each node is a multi dimensional Gaussian mixture model. And we can learn the GMM automatically from data and use BIC to select how many Gaussian in each node and the best structure at the same time.

Bibliography

[1] Bill Chiu, Eamonn Keogh, and Stefano Lonardi. Probabilistic discovery of time series motifs. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining KDD 03*, 304:493, 2003.

[2] Rónán Daly, Qiang Shen, and Stuart Aitken. Learning Bayesian networks: approaches and issues. *The Knowledge Engineering Review*, 26(02):99–157, jun 2011.

[3] G. Das, King-ip Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. *Knowledge Discovery and Data Mining*, pages 16–22, 1998.

[4] Dimitris Margaritis, Sebastian Thrun, Christos Faloutsos, Andrew W Moore, and Gregory F Cooper. Learning Bayesian Network Model Structure from Data. *Learning*, (May), 2003.

[5] Taketoshi Mori, Aritoki Takada, Hiroshi Noguchi, Tatsuya Harada, and Tomomasa Sato. Behavior prediction based on daily-life record database in distributed sensing space. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 1833–1839, 2005.

[6] Andrew A. Neath and Joseph E. Cavanaugh. The Bayesian information criterion: Background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2):199–203, 2012.

[7] Xavier Perrin, Francis Colas, Cedric Pradalier, Roland Siegwart, Ricardo Chavarriaga, and Jose del R. Millan. Learning User Habits for Semi-Autonomous\nNavigation Using Low Throughput Interfaces. pages 1–6, 2011.

[8] Lawrence Rabiner and Biing-Hwang Juang. An introduction to hidden Markov models. *ASSP Magazine, IEEE*, 3(January):4–16, 1986.

[9] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

[10] Dmitry Sinyukov, Ross Desmond, Matthew Dickerman, James Fleming, Jerome Schaufeld, and Taskin Padir. Multi-modal control framework for a semi-autonomous wheelchair using modular sensor designs. *Intelligent Service Robotics*, 7(3):145–155, 2014.

[11] Dmitry A Sinyukov, Ran Li, Nicholas W Otero, and Runzi Gao. Augmenting a Voice and Facial Expression Control of a Robotic Wheelchair with Assistive Navigation. *IEEE international Conference on Systems, Man, and Cybernetics*, pages 1088–1094, 2014.

[12] Padhraic Smyth and Rodney M. Goodman. An Information Theoretic Approach to Rule Induction from Databases, 1992.

[13] Thanh N. Tran, Klaudia Drab, and Michal Daszykowski. Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems*, 120:92–96, 2013.