1.204 Final Project

Implications from Clustering Mobility Patterns
- A New Look on Staggered Working Hours Strategy

Feifei Yu
December 11, 2012
# Contents

1. Introduction .................................................................................................................................2  
   1.1 Staggered Working Hours (SWH) ......................................................................................2  
   1.2 Principle Component Analysis in Activity Data .................................................................2  
2. Methodology ..................................................................................................................................3  
3. Data ...............................................................................................................................................4  
   3.1 Extracting Trip Chains ...........................................................................................................5  
   3.2 Generating TPM Combination Matrix ..................................................................................6  
   3.3 Generating TPM Binary Matrix ..............................................................................................8  
4. PCA Analysis for TPM ...................................................................................................................9  
5. Clustering of TPM Patterns .........................................................................................................11  
6. Implications for SWH Strategy .....................................................................................................16  
   6.1 Detection of Congestion .........................................................................................................16  
   6.2 Shift Plans for SWH Strategy ...............................................................................................17  
   6.3 Evaluation of Effects of SWH Strategy ................................................................................18  
7. Conclusion ....................................................................................................................................18  
8. References .....................................................................................................................................18  
9. Appendix - Code ............................................................................................................................19  
   9.1 Structure of MATLAB Code .................................................................................................19  
   9.2 “Trip_Chain_main.m” ............................................................................................................19  
   9.3 “generate_chain.m” ...............................................................................................................22  
   9.4 “TPM_Matrix.m” ...................................................................................................................22  
   9.5 “generate_binary.m” .............................................................................................................25  
   9.6 “create_C.m” ........................................................................................................................27  
   9.7 “draw_eigbehav.m” ...............................................................................................................28  
   9.8 “Construction_Error.m” .......................................................................................................32  
   9.9 “Clustering.m” ......................................................................................................................33  
   9.10 “Main.m” ............................................................................................................................37
1. Introduction

1.1 Staggered Working Hours (SWH)

Attached was the link for news from China Daily.
http://www.china.org.cn/china/2010-04/12/content_19793838.htm

As it described, “as many as 800,000 commuters in Beijing will change their rush hour schedules as the new adjusted-work-hour policy kicks off on Monday”. To be specific, several public organizations would “replace their former open hours from 8:30 am to 5:30 pm with the new working hours of 9 am to 6 pm”. This strategy would certainly ease Beijing’s Traffic congestion since Beijing is a city heavily relying on Subway and SWH strategy extends the rush hours.

Actually, this was not new to cities in Europe and North America and there were a variety of successful examples. The increasing application of SWH programs was mainly due to the following advantages.

First, the community benefits from SWH program since it encourages people to travel at varying times and therefore reduces traffic congestion and air pollution. Besides, it recognizes the employees’ demand to have flexible working schedules. Adding flexibility is important to accommodate family or leisure needs, especially for those female workers.

It seems that this is an “everyone-wins” strategy. But it has several obstacles for actual implementation. First, do we treat it to be voluntary or mandatory? Ideally, it is wise to offer as many shift options as possible so that workers can choose a schedule that is most convenient for them. But this is difficult to do in practice. Currently in China, most cities involved in this program only gives one mandatory choice and this lowers employees’ satisfaction. Second, and above all, what is the basis of applying staggered working hours? We need to know human’s clustering patterns in time dimension to decide the detailed strategy. In other words, if we detect same mobility patterns in certain period, people from those rush hours are willing to shift since it improves their life as well (easing bottlenecks in parking lots, office entrance, or even the elevators).

The following part will introduce the method for analyzing human mobility patterns.

1.2 Principle Component Analysis in Activity Data

The idea to apply PCA analysis to SWH strategy comes from two previous papers. One of them was from Eagle and Pentland (2009). They first introduced the notion
“Eigen-behaviors” to represent the behavioral structure of students at MIT through mobile phone logging locations. Then, Jiang (2012) applied “Eigen-behaviors” in Chicago Metropolitan Activity survey and get the individual’s daily activity structure. Besides, she partitioned individuals into clusters based on their daily activity.

These two papers provide techniques for analyzing temporal activity data and ways to cluster people without knowing demographics. I will implement the techniques in my activity survey data from Shangyu and exact implications for making Staggered Working Hours (SWH) strategies.

2. Methodology

The aim of this analysis is to detect clustering patterns for travels of different modes in time dimension and to make SWH recommendations based on the temporal conflict. Therefore, instead of an “Activity Matrix” as in Eagle and Jiang’s paper, I create a Trip Purpose-Mode Combination (I will call it “TPM” in the rest of this report) matrix to get the information needed for SWH analysis.

With TPM matrix (Dimension: samples * Time_Intervals_Per_Day), we can perform the following procedures to extract eigenvectors (“Eigen_TPM”) and their related eigenvalues.

1. Generate TPM Binary matrix for each TPM type (Dimension: samples * Time_Intervals_Per_Day* TPM _Types)
2. Subtract the mean from TPM Binary matrix
3. Calculate the variance-covariance matrix
4. Calculate the eigenvectors and eigenvalues of covariance matrix

Then, it requires us to choose “k” components with first “k” largest eigenvalues to form a feature vector “FV”. This feature vector takes eigenvectors that you choose and is formed with these eigenvectors in columns.

\[ FV = (eig_1, eig_2, \ldots, eig_k) \]

With feature vector, it is convenient to reconstruct the TPM Binary matrix from the projection of original samples on eigenvectors and the mean we subtracted in step (2). One thing to notice here is that we will not get the exactly the same original matrix back if we do not take all the eigenvectors. Actually in here, we calculate the reconstruction error based on reconstructed TPM Binary matrix and select a reasonable number of eigenvectors to conduct the dimension reduction. With that reduced-dimension matrix (Dimension: samples *k), we are able to perform the K-means techniques with less computational challenge and cluster people based on
their trip-mode combination patterns.

For K-means clustering method, I use silhouette index for the diagnostics. If the silhouette index is 1, it means we get the perfect cluster for certain points. If it is 0, it indicates that this point is not distinguishable for one cluster or another. If the index is -1, the cluster is probably wrong for the current point. Therefore, we will select proper number of groups with relatively larger silhouette index (positive, of course).

3. Data

The city selected in the research is Shangyu, a typical small city located in the hinterland of Yangtze River Delta, one of the fastest growing regions in the east of China. The city has a population of about 204,900 (average town population in china is 930,475) and it covers an area of 111 square kilometers with a central district of 18.2 square kilometers. Shangyu belongs to the first opening coastal cities in China.

The data is collected in the form of questionnaires, using the method of random sampling (sampling rate is approximately 3%) and face-to-face interviews. The survey was conducted on April, 2006 and the sample consists of 6253 individuals for their one-day activity records. The survey itself includes three sections: household characteristics, individual socio-demographics, and travel-activity attributes.

In order to perform PCA analysis, we need to transfer the original data structure into a Trip-Mode combination matrix. The original database is like the following:

<table>
<thead>
<tr>
<th>PID</th>
<th>Start Time</th>
<th>Purpose</th>
<th>Mode</th>
<th>Arrive Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7:20</td>
<td>1</td>
<td>4</td>
<td>7:30</td>
</tr>
<tr>
<td>1</td>
<td>11:00</td>
<td>7</td>
<td>4</td>
<td>11:10</td>
</tr>
<tr>
<td>1</td>
<td>13:20</td>
<td>1</td>
<td>2</td>
<td>13:40</td>
</tr>
<tr>
<td>1</td>
<td>17:50</td>
<td>7</td>
<td>2</td>
<td>18:00</td>
</tr>
<tr>
<td>2</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Table 1 – Original data structure

Here the “PID” refers to the person ID and each person has multi-line records. The trip purposes are categorized into 9 types: 1) Work, 2) School, 3) Official Business, 4) Shopping, 5) Leisure, 6) Picking-up Others, 7) Back-home, 8) Back-to-office, and 9)
others. The modes also have 9 categories, including 1) Walk, 2) Bike or E-bike, 3) Pedicab, 4) Transit, 5) Taxi, 6) Motorcycle, 7) Private car, 8) Company car, and 9) Others.

The rest of this chapter discusses in detail for the data processing.

3.1 Extracting Trip Chains

A trip chain, or tour, is defined as the travel from home to one or more activity locations and back home again. For each individual, the trip chain depicts his/her activity schedule in an entire day. Since it is home-based, it should start from home and go back home for the last trip. I use this criterion to exclude non-home based chains and the summary statistics are shown as follows.

![Figure 1 – Distribution of trip chains with frequencies more than 1%](image)

Figure 1 gives 10 types of trip chains, each of which accounts for at least 1 percent of the total chains. We can see that the 4 dominate trip chain types are: 17 (Home–Work–Home), 1717 (Home-Work-Home-Work-Home), 27 (Home–School–Home), 47 (Home-Shopping-Home). Most of chains are short with only 2 or 4 trips per day.

Trip chains give us general information about individual’s daily activity schedules. It also serves as a tool for us to exclude unreasonable samples with non-home based trip chains and trip chains that last more than 24 hours (impossible to analyze in PCA).

Please find the detailed code for this part in appendix.
3.2 Generating TPM Combination Matrix

Originally, we have 9 types of trip purposes and 9 categories of modes. It is unrealistic to analyze $9 \times 9 = 81$ combinations. Instead, I aggregate the purposes and modes as follows.

<table>
<thead>
<tr>
<th>Trip Categories</th>
<th>Purposes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. School Trips</td>
<td>School, Back-to-Home-from-School</td>
</tr>
<tr>
<td>5. Others</td>
<td>Others</td>
</tr>
</tbody>
</table>

Table 2 – Aggregated 5 types of purposes

<table>
<thead>
<tr>
<th>Mode Categories</th>
<th>Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Walk</td>
<td>Walk</td>
</tr>
<tr>
<td>2. Non-motorized-lane modes</td>
<td>Bike or E-bike, Pedi-cab</td>
</tr>
<tr>
<td>3. Public Transit</td>
<td>Transit</td>
</tr>
<tr>
<td>4. Cars &amp; Motors</td>
<td>Motorcycle, Private Car, Company Car</td>
</tr>
<tr>
<td>5. Taxi</td>
<td>Taxi</td>
</tr>
<tr>
<td>6. Others</td>
<td>Others</td>
</tr>
</tbody>
</table>

Table 3 – Aggregated 6 types of modes

This aggregation reduces 81 combinations to 30 combinations. Still, we cannot do the PCA analysis with so many categories. Therefore, I did the frequency analysis for those 30 combinations.

Figure 2 – Cumulative percentage for the major types of TPM combinations
I code 30 combinations using the following rules: (Trip_Code – 1)*6 + Mode_Code. For example, School Trips (2) and Transit (3) combination will be coded as “9”.

We can observe from figure 2 that the listed 9 types of TPM combinations already account for more than 85% of the data. So it is a reasonable number of types we want to incorporate for the following PCA and clustering analysis. The detailed information along with static state is described in table 4.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Modes</th>
</tr>
</thead>
<tbody>
<tr>
<td>School</td>
<td>Non-motorized-lane modes</td>
</tr>
<tr>
<td>School</td>
<td>Walk</td>
</tr>
<tr>
<td>Subs</td>
<td>Walk</td>
</tr>
<tr>
<td>Leis</td>
<td>Walk</td>
</tr>
<tr>
<td>Maint</td>
<td>Non-motorized-lane modes</td>
</tr>
<tr>
<td>Maint</td>
<td>Walk</td>
</tr>
<tr>
<td>Subs</td>
<td>Motorcycle, Private Car, Company Car</td>
</tr>
<tr>
<td>Subs</td>
<td>Non-motorized-lane modes</td>
</tr>
<tr>
<td>Other</td>
<td>Non-motorized-lane modes</td>
</tr>
<tr>
<td>Static</td>
<td></td>
</tr>
</tbody>
</table>

Table 4 – Selected 10 types of TPM combinations (including static)

I use trips instead of activities because we want to know exactly when the person starts and ends the travel. This is particularly important for mobility analysis. With those 10 selected TPM combinations, we can create TPM matrix for each 10 minutes time interval (144 in total for 24 hours) and visualize it to show the general picture in terms of temporal characteristics for individual’s assorted trip purposes and modes.
Figure 3 – Individual trip purpose & mode variation across the day

Figure 3 describes the 24-hour human trip purpose & mode use variations. The x axis represents the time of the day and the y axis is for the sample ID. We can detect two clear peak hours here from 7am to 8am in the morning and from 17pm to 18pm in the evening. The trips are concentrated around noon as well for two periods (from 11am to 12pm and from 13pm to 14pm).

3.3 Generating TPM Binary Matrix

After data cleaning and purpose & mode combination selection, the sample size now is 4018 individuals. The dimension of TPM matrix is 4018*144 (10 minutes interval). Then we can generate a binary matrix for all the 10 types of TPM combinations (including static). The dimension of this binary matrix becomes 4018*1440 and each sample get a vector of 0s and 1s for each trip purpose & mode combination.

Figure 4 – Binary visualization for each individual trip & mode combination
4. PCA for TPM

Using the PCA that we discussed above, we can now draw first three eigenvectors with largest eigenvalues.

From figure 5, the first eigenvector shows the high probability of staying at home before 8am and then conducting maintenance trips with walk. The second eigenvector describes high probability of travelling to work right before 8am, leaving office at noon, back-to offices 2 hours later and finally returning home at around 17pm with non-motorized-lane modes. The third eigenvector reveals the same time use patterns as in second eigenvector, except it is for cars or motors instead of non-motorized-lane modes.

![First three eigenvectors for individual trip & mode combinations](image)

The first three eigenvectors only account for three largest variances of individual’s TPM combinations. This is not enough for Staggered Working Hours Strategy.

![Reconstructed Binary TPM matrix with first three eigenvectors](image)
Figure 6 shows the reconstructed binary TPM matrix with these three eigenvectors. Compared with figure 4, we can observe that those three eigenvectors only reconstruct subsistence trips but totally ignore other types like maintenance and school trips. Therefore, we need more information to reconstruct the binary TPM matrix with a high precision.

Figure 7 presents the trend of eigenvalues with rank of eigenvectors in the left chart and the reconstruction error with the increasing number of eigenvectors in the right. We can observe a quick drop for eigenvalues in the beginning. As for the error rate, Jiang uses 1% as threshold but things are different here.

By achieving 1% overall reconstruction error, we need only 3 eigenvectors. But as is shown above, the binary TPM matrix loses its precision in non-subistence trips. This shows an important fact. Although we can achieve a relatively low average error, it does not always imply the success of reconstruction. It might be true for the major trip types like subsistence trips. But it fails for other types if we are still interested in other types (we want to detect the collision of travel in time dimension for different types of trips). Even with 20 eigenvectors (error rate less than 0.7%), we should still reject this reconstruction since it loses information for maintenance and school non-motorized-lane modes trips (see in figure 8).
When the number of eigenvectors increases to 30, we are able to obtain a good reconstruction both with low average error rate (less than 0.6%) and comprehensive coverage for all types of trips (see in figure 9).

![Figure 9 – Reconstructed Binary TPM matrix with first 30 eigenvectors](image)

### 5. Clustering of TPM Patterns

If we use the normal Euclidean distance between two individuals, the calculation for distance is computationally challenging (individual vector dimension: 1*1440). Therefore, we need PCA technique to reduce dimension but without losing too much information. As is shown above, reconstruction with first 30 eigenvectors gives a good representation of original TPM binary matrix. Therefore, we apply the K-Means method here to partition individuals into different TPM groups with reconstructed data (by 30 eigenvectors).

Another important issue is to determine the proper number of groups for clustering. I use the mean Silhouette index here for the diagnostics of clustering.

Figure 10 shows the change of Silhouette index with the increased number of cluster groups. I don’t want the group number to be too small so that we cannot distinguish clearly for the purpose of implementing Staggered Working Hours program. But in the meantime I need to keep the index relatively high. So I will choose 6 as the group number for clustering.
Figure 10 – Diagnostic index for K-Means Clustering

Figure 11 to 16 show the TPM combination matrix for the 6 clustered groups. Then I calculate demographic characteristics within each group in table 5.

Figure 11 – TPM matrix for group 1
Figure 12 – TPM matrix for group 2

Figure 13 – TPM matrix for group 3

Figure 14 – TPM matrix for group 4
Figure 15 – TPM matrix for group 5

Figure 16 – TPM matrix for group 6

<table>
<thead>
<tr>
<th>Demographics</th>
<th>Grand Mean</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Female</td>
<td>49.5%</td>
<td>69.8%</td>
<td>23.0%</td>
<td>52.9%</td>
<td>63.3%</td>
<td>50.6%</td>
<td>38.7%</td>
</tr>
<tr>
<td>workers</td>
<td>64.4%</td>
<td>11.1%</td>
<td>96.2%</td>
<td>97.7%</td>
<td>95.0%</td>
<td>42.2%</td>
<td>89.5%</td>
</tr>
<tr>
<td>students</td>
<td>15.5%</td>
<td>18.1%</td>
<td>3.6%</td>
<td>0.2%</td>
<td>1.6%</td>
<td>37.4%</td>
<td>8.1%</td>
</tr>
<tr>
<td>Homemaker</td>
<td>7.8%</td>
<td>30.5%</td>
<td>0.0%</td>
<td>0.2%</td>
<td>1.6%</td>
<td>6.7%</td>
<td>1.1%</td>
</tr>
<tr>
<td>Retired</td>
<td>12.3%</td>
<td>40.3%</td>
<td>0.2%</td>
<td>1.9%</td>
<td>1.8%</td>
<td>13.7%</td>
<td>1.3%</td>
</tr>
<tr>
<td>Young (age&lt;30)</td>
<td>23.7%</td>
<td>3.6%</td>
<td>10.8%</td>
<td>14.0%</td>
<td>14.5%</td>
<td>42.7%</td>
<td>19.0%</td>
</tr>
<tr>
<td>Elder (age&gt;60)</td>
<td>11.6%</td>
<td>34.4%</td>
<td>2.4%</td>
<td>2.4%</td>
<td>2.6%</td>
<td>13.6%</td>
<td>2.6%</td>
</tr>
<tr>
<td>Edu ≥ College</td>
<td>19.4%</td>
<td>6.1%</td>
<td>45.0%</td>
<td>30.7%</td>
<td>14.5%</td>
<td>11.1%</td>
<td>25.2%</td>
</tr>
<tr>
<td>Driving license</td>
<td>19.8%</td>
<td>7.9%</td>
<td>58.6%</td>
<td>15.3%</td>
<td>11.1%</td>
<td>11.3%</td>
<td>25.8%</td>
</tr>
<tr>
<td>Low income</td>
<td>6.0%</td>
<td>10.3%</td>
<td>1.5%</td>
<td>3.1%</td>
<td>4.7%</td>
<td>6.7%</td>
<td>5.1%</td>
</tr>
<tr>
<td>High income</td>
<td>32.3%</td>
<td>24.0%</td>
<td>51.9%</td>
<td>34.2%</td>
<td>21.4%</td>
<td>29.2%</td>
<td>30.6%</td>
</tr>
</tbody>
</table>

Table 5 – Demographics for 6 clustered groups
With figure 11-16 and table 5, we can have a good look at the clustering mobility patterns for all the 6 groups.

Group 1 have a relatively high proportion of female, homemaker, and elder retired individuals. They usually walk to leisure from 6am to 8am in the early morning and take non-motorized-lane modes to maintenance afterwards from 8am to 10am. Part of them might also go out for maintenance from 2pm to 4pm in the afternoon with non-motorized-lane modes and conduct several leisure trips on foot as well. This group consists of 17.6% of the samples. The ownership of driving license is only 7.9%, which is much lower than the grand mean (19.8%).

Group 2 is featured for a high proportion of male, workers, and driving license ownership. The temporal patterns include the following sequence: 1) travel to work from 7:30am to 8am in the morning, 2) go back home to have a break from 11am to 12pm in the noon, 3) go back to office from 1pm to 2pm in the early afternoon, and 4) return home after work from 17pm to 18pm in the late afternoon. This sequence is conducted by motorized-modes as cars or motorcycles. There are 551 samples (13.7%) belonging to this group.

For group 3, more than 97% of them are workers. Other sample demographics within group are pretty close to the average across all groups. The clustering mobility patterns are similar to those in group 3, except that the trips are taken by non-motorized-lane modes. There might be some maintenance trips after work in the evening with non-motorized-lane modes. This group accounts for 13.3% of the total sample.

Group 4 is composed of similar demographic groups as group 3. It also shares the similar travel patterns in the morning and late afternoon. But the back-to-office trip and second back-home trip are no longer concentrated. We have 9.6% of the sample clustered in this group.

Group 5 incorporate typical travel flows for students from 6am to 8pm in the morning, and from 16pm to 18pm in the late afternoon. The modes are either walk or non-motorized-lane modes. Also, several morning and evening leisure trips are involved with walk. The sample size here is 1304 (32.4%).

Group 6 is typical for 2 motorized-mode work trips per day – from 7am to 8am in the morning, and from 16pm to 18pm in the late afternoon. The sample has a size of 532 (13.2%) and a relatively high proportion for male and workers.

In general, we can detect clear temporal patterns in terms of travel purpose and mode use within each clustered group. This is of particular interest for the applying of SWH strategies.
6. Implications for SWH Strategy

The clustered mobility patterns have important implications for Staggered Working Hours Strategy. It can help the policy maker to detect congestion, make shifts based on temporal conflict, and evaluate potential effects of SWH strategy.

6.1 Detection of Congestion

The road is usually divided into sidewalk, non-motorized-lane, and drive way. For the modes defined in this research, walk trips are conducted in sidewalk, non-motorized-lane modes are taken in non-motorized-lane, and public transit, taxi, and cars & motors uses the driveway. The assumption here is that the congestion only happens between modes in the same type of lane at the same time. With this principle, we are able to detect several congestion types for the 6 clustered mobility patterns.

(1) Congestion in non-motorized-lanes

Diagram 1:

We can observe congestion among subsistence, maintenance, and school trips in the morning and evening peak for non-motorized-lanes modes.

(2) Congestion in motorized-lanes

Diagram 2:

We can detect congestion for motorized-subistence trips in the morning and evening peak.
6.2 Shift Plans for SWH Strategy

In order to make plans for SWH strategy, we need both the diagrams from 6.1 and the population information for all occupations. Table 6 lists the occupation share within 6 clustered mobility groups.

<table>
<thead>
<tr>
<th>Occupations</th>
<th>Proportion within each cluster</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td></td>
<td>0.8%</td>
<td>0.5%</td>
<td>0.2%</td>
<td>1.6%</td>
<td>37.4%</td>
<td>8.1%</td>
</tr>
<tr>
<td>Manufacturing Industrial worker</td>
<td></td>
<td>2.0%</td>
<td>13.8%</td>
<td>15.1%</td>
<td>23.8%</td>
<td>5.4%</td>
<td>20.9%</td>
</tr>
<tr>
<td>Service Industrial worker</td>
<td></td>
<td>3.5%</td>
<td>5.4%</td>
<td>14.4%</td>
<td>22.7%</td>
<td>5.1%</td>
<td>9.8%</td>
</tr>
<tr>
<td>Government, public organization worker</td>
<td></td>
<td>5.1%</td>
<td>52.8%</td>
<td>48.8%</td>
<td>25.3%</td>
<td>12.7%</td>
<td>30.1%</td>
</tr>
<tr>
<td>Self-employed worker</td>
<td></td>
<td>11.1%</td>
<td>15.6%</td>
<td>6.9%</td>
<td>11.1%</td>
<td>11.4%</td>
<td>19.5%</td>
</tr>
<tr>
<td>Retired individual</td>
<td></td>
<td>40.3%</td>
<td>1.8%</td>
<td>1.9%</td>
<td>1.8%</td>
<td>13.7%</td>
<td>1.3%</td>
</tr>
<tr>
<td>Homemaker</td>
<td></td>
<td>30.5%</td>
<td>0.0%</td>
<td>0.2%</td>
<td>1.6%</td>
<td>6.7%</td>
<td>1.1%</td>
</tr>
<tr>
<td>Farmer</td>
<td></td>
<td>1.1%</td>
<td>0.2%</td>
<td>0.9%</td>
<td>0.3%</td>
<td>1.4%</td>
<td>0.9%</td>
</tr>
<tr>
<td>Others</td>
<td></td>
<td>5.5%</td>
<td>9.8%</td>
<td>11.6%</td>
<td>11.9%</td>
<td>6.1%</td>
<td>8.3%</td>
</tr>
</tbody>
</table>

Table 6 – Occupation share for 6 clustered groups

Students and Government, public organization workers are the group easier to implement SWH policy. Thus I will focus on those two groups when making strategies. Diagram 3 and 4 give one possible example of work hour shifts.

Diagram 3:

Diagram 4:
6.3 Evaluation of Effects of SWH Strategy

If we apply the strategy stated in diagram 3 and 4 for Government, public organization workers, we need to know the potential effect of implementing the policy. The following steps show how we can have a rough estimation for the ease of congestion level.

1) In chapter 3.2, we select 9 main types of TPM from all 30 categories. We capture the information for 85.9% trips.

2) For non-motorized lanes, we avoid 3/4 of conflict for group 3 (13.3% of total sample) and 3/4 of conflict for group 4 (9.6% of total sample). With the occupation share data, we can calculate the ease of congestion in non-motorized lane:

\[
85.9\% \times \left( 13.3\% \times \frac{3}{4} \times 48.8\% + 9.6\% \times \frac{3}{4} \times 25.3\% \right) = 5.7\%
\]

3) For non-motorized lanes, we avoid \((1/2 \times 1/2 + 1/2 \times 2/3 = 7/12)\) of conflict for group 6 (13.2% of total sample). With the occupation share data, we can calculate the ease of congestion in motorized lane:

\[
85.9\% \times 13.2\% \times \frac{7}{12} \times 30.1\% = 2.0\%
\]

The above example is just a naive demo of the idea. But it shows great potential to combine PCA, K-Means techniques with real world SWH policy making process.

7. Conclusion

This research applies PCA and Clustering techniques for activity survey data in Shangyu. Instead of constructing the activity matrix, I create a trip purpose-mode (TPM) combination matrix to extract the eigenvectors and partition individuals into 6 mobility groups. With the clustered groups and demographics within each group, I am able to put forward a SWH strategy and evaluate the effect in terms of travel congestion. This example is pretty naive and the work is actually an exploration of this new area. Much improvement could be made with better data by adding the location factors into account. In a word, this type of analysis shows potential for SWH policy making and it’s easy to promote since the activity data is often available for the purpose of transportation planning survey.

8. References

9. Appendix - Code

9.1 Structure of MATLAB Code

“Trip_Chain_main.m” – Read the data, extracts trip chains, and excludes unreasonable samples
“generate_chain.m” – Function to create trip chain given sequence of trips
“TPM_Matrix.m” – Generate trip purpose & mode (TPM) matrix (similar as “color matrix”)  
“generate_binary.m” – Function to generate binary matrix from TPM matrix
“create_C.m” – Function to create variance-covariance matrix
“draw_eigbehav.m” – Function to draw eigenvector
“Construction_Error.m” – function to get the reconstructed matrix and calculate the error rate
“Clustering.m” – Perform K-Means clustering analysis
“Main.m” – Draw the plots for all parts

9.2 “Trip_Chain_main.m”

```matlab
% Extracting trip chains
% Demo: ID, Start Time, End Time, Purpose, Mode

% Read the data
data = csvread('Demo.csv');
rows = size(data,1);
cols = size(data,2);

% D matrix with ID, Start Time, End Time, Purpose, Mode, 
% Aggregated purpose, Aggregated mode, Purpose & Mode combination
D = zeros(rows,cols+3);

% Re-range the ID
ID = 1;
for t=1:(rows-1)
    if(data(t,1)==data(t+1,1))
        D(t,1) = ID;
    else
        D(t,1) = ID;
        ID = ID + 1;
    end
end
D(rows,1) = ID;

% Form matrix D
D(:,2) = data(:,2);
D(:,3) = data(:,3);
D(:,4) = data(:,4);
D(:,5) = data(:,5);
```
% Aggregate Trip Purposes
for i=1:rows
    if (D(i,4)==1 || D(i,4)==3 || D(i,4)==8)
        D(i,6) = 1;
    elseif (D(i,4)==2)
        D(i,6) = 2;
    elseif (D(i,4)==4 || D(i,4)==6)
        D(i,6) = 3;
    elseif (D(i,4)==5)
        D(i,6) = 4;
    elseif (D(i,4)==9)
        D(i,6) = 5;
    elseif (D(i,4)==7) % For back-home purposes
        D(i,6) = 7;
    else
        D(i,6) = 7;
    end
end
% Dealing with back-home purposes
for i=2:rows
    if (D(i,6)==7 && D(i-1,6)==1)
        D(i,6) = 1;
    elseif (D(i,6)==7 && D(i-1,6)==2)
        D(i,6) = 2;
    elseif (D(i,6)==7 && D(i-1,6)==3)
        D(i,6) = 3;
    elseif (D(i,6)==7 && D(i-1,6)==4)
        D(i,6) = 4;
    elseif (D(i,6)==7 && D(i-1,6)==5)
        D(i,6) = 5;
    else
        D(i,6) = 7;
    end
end
% Aggregate Modes
for i=1:rows
    if (D(i,5)==1)
        D(i,7) = 1;
    elseif (D(i,5)==2 || D(i,5)==3)
        D(i,7) = 2;
    elseif (D(i,5)==4)
        D(i,7) = 3;
    elseif (D(i,5)==6 || D(i,5)==7 || D(i,5)==8)
        D(i,7) = 4;
    elseif (D(i,5)==9)
        D(i,7) = 5;
    else
        D(i,7) = 5;
    end
end
% Purpose & Mode combination
D(:,8) = (D(:,6)-1)*6 + D(:,7);

% Using matrix D to extract trip chains
Chain = zeros(1,10);
Chain_Index = 1;
persons = max(D(:,1));
Trip_Chain = zeros(persons,1);
PID = 1;
for i=1:(rows-1)
  if (D(i,1)==D(i+1,1))
    Chain(Chain_Index) = D(i,4);
    Chain_Index = Chain_Index + 1;
  else
    Chain(Chain_Index) = D(i,4);
    Trip_Chain(PID) = generate_chain(Chain,Chain_Index);
    Chain = zeros(1,10);
    Chain_Index = 1;
    PID = PID + 1;
  end
end
Chain(Chain_Index) = D(i+1,4);
Trip_Chain(PID) = generate_chain(Chain,Chain_Index);

% ID match matrix (column 1: original ID; column 2: new ID)
Original_ID = unique(data(:,1));
New_ID = unique(D(:,1));
ID_match = [Original_ID,New_ID];

% Generate cleaned trip chain matrix by excluding unreasonable chains
Trip_Chain_Clean = [Original_ID,New_ID,Trip_Chain];

% Exclude non home-based trip chains
Delete_Type1 = zeros(500,1); % may change the size here
Delete_Count_Type1 = 1;
for j=1:PID
  Last_Trip = mod(Trip_Chain_Clean(j,3),10);
  if (Last_Trip==7)
    Delete_Type1(Delete_Count_Type1)=j;
    Delete_Count_Type1 = Delete_Count_Type1 + 1;
  end
end
end
Delete_Index_Type1 = Delete_Type1(Delete_Type1>0);

% Exclude trip chains lasting more than 24 hours
Delete_Type2 = zeros(500,1); % may change the size here
Delete_Count_Type2 = 1;
for k=1:rows-1
    if(D(k,3)<=D(k,2))
        Delete_Type2(Delete_Count_Type2)=D(k,1);
        Delete_Count_Type2 = Delete_Count_Type2 + 1;
    elseif(D(k,1)==D(k+1,1) && D(k,2)>D(k+1,2))
        Delete_Type2(Delete_Count_Type2)=D(k,1);
        Delete_Count_Type2 = Delete_Count_Type2 + 1;
    end
end
Delete_Index_Type2 = Delete_Type2(Delete_Type2>0);
Delete_Index_Type2 = unique(Delete_Index_Type2);

Delete_Index_Temp=[Delete_Index_Type1;Delete_Index_Type2];
Delete_Index=unique(Delete_Index_Temp);
Trip_Chain_Clean(Delete_Index,:)=[];

9.3 “generate_chain.m”

% Generate trip chain for each individual

function chain_out = generate_chain(Chain,Chain_Length)

    chain_out = 0;
    Index = Chain_Length;

    for k=1:Chain_Length
        chain_out = chain_out + Chain(Index)*10^(k-1);
        Index = Index-1;
    end

9.4 “TPM_Matrix.m”

% Trip Purpose & Mode (TPM) combination matrix

% Screen out unreasonable records
D_Temp = D;
Delete_num = length(Delete_Index);
for i=1:Delete_num
    ind = find(D_Temp(:,1)==Delete_Index(i));
    D_Temp(ind,:) = [];
end

% Go back to SPSS and do the frequency analysis
% The results are summarized as follows
% We will choose 13 TPM types and re-code them
% 1: Subsistence Non-Motor (2)
% 2: Subsistence Cars & Motors (4)
% 3: Maintenance Walk (13)
% 4: Maintenance Non-motor (14)
% 5: Leisure Walk (19)
% 6: Subsistence Walk (1)
% 7: School Walk (7)
% 8: School Non-motor (8)
% 9: Others Non-Motor(26)

% Select out aggregated TPMs
for i=1:length(D_Temp(:,1))
    if(D_Temp(i,8)==2)
        D_Temp(i,9)= 1;
    elseif(D_Temp(i,8)==4)
        D_Temp(i,9)= 2;
    elseif(D_Temp(i,8)==13)
        D_Temp(i,9)= 3;
    elseif(D_Temp(i,8)==14)
        D_Temp(i,9)= 4;
    elseif(D_Temp(i,8)==19)
        D_Temp(i,9)= 5;
    elseif(D_Temp(i,8)==1)
        D_Temp(i,9)= 6;
    elseif(D_Temp(i,8)==7)
        D_Temp(i,9)= 7;
    elseif(D_Temp(i,8)==8)
        D_Temp(i,9)= 8;
    elseif(D_Temp(i,8)==26)
        D_Temp(i,9)= 9;
    else
        D_Temp(i,9)= 0;
    end
end
D_Temp2 = D_Temp(D_Temp(:,9)>0,:);
rows_new = size(D_Temp2,1);
cols_new = size(D_Temp2,2);
D_Final = zeros(rows_new,cols_new);

% Re-range the ID (the second time)
ID = 1;
for t=1:(rows_new-1)
    if (D_Temp2(t,1)==D_Temp2(t+1,1))
        D_Final(t,1) = ID;
    else
        D_Final(t,1) = ID;
        ID = ID + 1;
    end
end
D_Final(rows_new,1) = ID;

% Form matrix D_Final
D_Final(:,2) = D_Temp2(:,2);
D_Final(:,3) = D_Temp2(:,3);
D_Final(:,4) = D_Temp2(:,4);
D_Final(:,5) = D_Temp2(:,5);
D_Final(:,6) = D_Temp2(:,6);
D_Final(:,7) = D_Temp2(:,7);
D_Final(:,8) = D_Temp2(:,8);
D_Final(:,9) = D_Temp2(:,9);

% ID match matrix 2 (column 1: original ID; column 2: new ID)
Original_ID_2 = unique(D_Temp2(:,1));
New_ID_2 = unique(D_Final(:,1));
ID_match_2 = [Original_ID_2,New_ID_2];

% Initialization
persons_Final = length(unique(D_Final(:,1)));
period = 24*6; % 10 minutes interval
TPM = zeros(persons_Final,period);
Person_ID = 1;

% Generate TPM matrix
for i=1:size(D_Final,1)-1
    if (D_Final(i+1,1)==D_Final(i,1))
        start_T = floor(D_Final(i,2)/10);
        end_T = floor(D_Final(i,3)/10);
        TPM(Person_ID,start_T:end_T)=D_Final(i,9);
    else
        start_T = floor(D_Final(i,2)/10);
        end_T = floor(D_Final(i,3)/10);
TPM(Person_ID,start_T:end_T)=D_Final(i,9);
Person_ID = Person_ID + 1;
end
end
i = i + 1;
start_T = floor(D_Final(i,2)/10);
end_T = floor(D_Final(i,3)/10);
TPM(Person_ID,start_T:end_T)=D_Final(i,9);

9.5 “generate_binary.m”

function [Mbw] = generate_binary(Mcolor)

    % hours=size(Mcolor,1);
    % days=size(Mcolor,2);
    % num_labels=5;
    % Mbw = zeros(days,num_labels*hours);
    
    % for i = 1:days
    %     for j = 1:hours
    %         place=Mcolor(j,i);
    %         if(isnan(place))
    %             Ji=hours*(num_labels-1)+1; %97=24*(5-1)+1 for off
    %             Jf=hours*num_labels; %120
    %             J=Ji+j-1;
    %             Mbw(i,J)=1;
    %         else
    %             if(place==0)
    %                 Ji=hours*(num_labels-2)+1; %73 for no signal
    %                 Jf=hours*(num_labels-1); %96
    %                 J=Ji+j-1;
    %                 Mbw(i,J)=81;
    %             else
    %                 Ji=hours*(place-1)+1; % (1-24) for house and (25-48) for work
    %                 Jf=hours*place; % and (49-72) for elsewhere
    %                 J=Ji+j-1;
    %                 Mbw(i,J)=1;
    %             end
    %         end
    %     end
    % end

samples=size(Mcolor,1);
time=size(Mcolor,2);
um_labels=10;
Mbw = zeros(samples,num_labels*time);

for i = 1:samples
    for j = 1:time
        TPM_Type=Mcolor(i,j);
        if(TPM_Type==9)
            Ji=time*(num_labels-1)+1; %144*(10-1)+1 for Subs_NM
            Jf=time*num_labels; %144*10
            J=Ji+j-1;
            Mbw(i,J)=1;
        else
            if(TPM_Type==8)
                Ji=time*(num_labels-2)+1; %144*(10-2)+1
                Jf=time*(num_labels-1); %144*9
                J=Ji+j-1;
                Mbw(i,J)=1;
            else
                if(TPM_Type==7)
                    Ji=time*(num_labels-3)+1; %144*(10-3)+1
                    Jf=time*(num_labels-2); %144*8
                    J=Ji+j-1;
                    Mbw(i,J)=1;
                else
                    if(TPM_Type==6)
                        Ji=time*(num_labels-4)+1; %144*(10-4)+1
                        Jf=time*(num_labels-3); %144*7
                        J=Ji+j-1;
                        Mbw(i,J)=1;
                    else
                        if(TPM_Type==5)
                            Ji=time*(num_labels-5)+1; %144*(10-5)+1
                            Jf=time*(num_labels-4); %144*6
                            J=Ji+j-1;
                            Mbw(i,J)=1;
                        else
                            if(TPM_Type==4)
                                Ji=time*(num_labels-6)+1; %144*(10-6)+1
                                Jf=time*(num_labels-5); %144*5
                                J=Ji+j-1;
                                Mbw(i,J)=1;
                            else
                                if(TPM_Type==3)
                                    Ji=time*(num_labels-7)+1; %144*(10-7)+1
                                    Jf=time*(num_labels-6); %144*4
                                    J=Ji+j-1;
                                    Mbw(i,J)=1;
else
    if (TPM_Type==2)
        Ji=time*(num_labels-8)+1; 144*2+1
        Jf=time*(num_labels-7); 144*3
        J=Ji+j-1;
        Mbw(i,J)=1;
    else

        if (TPM_Type==1)
            Ji=time*(num_labels-9)+1; 144*1+1
            Jf=time*(num_labels-8); 144*2
            J=Ji+j-1;
            Mbw(i,J)=1;
        else
            Ji=time*TPM_Type+1; 1
            Jf=time*(TPM_Type+1); 144
            J=Ji+j-1;
            Mbw(i,J)=1;
        end
    end
end
end
end
end
end
end

9.6 “create_C.m”

function [C] = create_C(MBW)

% H=size(MBW,2);
% Days=size(MBW,1);
% Psi = mean(MBW,1);
% for i =1:Days
%   Gamma(i,:)=MBW(i,:)-Psi;
%   A(i,:)=Gamma(i,:);
% end
% C=A'*A;

samples=size(MBW,1);
Psi = mean(MBW,1);
Gamma = zeros(size(MBW,1),size(MBW,2));
A = zeros(size(MBW,1),size(MBW,2));

for i =1:samples
    Gamma(i,:)=MBW(i,:)-Psi;
    A(i,:)=Gamma(i,:);
end
C=A'*A;

9.7 "draw_eigbehav.m"

function draw_eigbehav(V,eig_index)

% num_labels = 5;
% hours=24;
%
% house_i=1;
% work_i=2;
% elsewhere_i=3;
% nosig=4;
%
% figure;
% colormap hot;
% Ji=hours*(house_i-1)+1; %indexes of house 1-24
% Jf=hours*house_i;
% subplot(4,1,1)
% imagesc(V(Ji:Jf,eig_index)) %drawing the first components of the given index
% title('Home','FontSize',16)
% caxis([-0.05 0.15])
% colorbar;
%
% figure;
% colormap hot;
% Ji=hours*(work_i-1)+1; %indexes of work 25-48
% Jf=hours*work_i;
% subplot(4,1,2)
% imagesc(V(Ji:Jf,eig_index))
% colorbar;
% caxis([-0.05 0.15])
% title('Work','FontSize',16)
%
% figure;
% colormap hot;
% Ji=hours*(elsewhere_i-1)+1; %indexes of elsewhere 49-72
% Jf=hours*elsewhere_i;
% subplot(4,1,3)
% imagesc(V(Ji:Jf,eig_index))
% colorbar;
% caxis([-0.05 0.15])
% title('Elsewhere','FontSize',16)
%
% figure;
% colormap hot;
% Ji=hours*(nosig-1)+1;
% Jf=hours*nosig; %indexes of no signal 73-96
% subplot(4,1,4)
% imagesc(V(Ji:Jf,eig_index)')
% title('NoSig','FontSize',16)
% caxis([-0.05 0.15])
% colorbar;

% num_labels = 10;
time=144;

Static_i=0;
Subs_NM_i=1;
Subs_CM_i=2;
Main_WA_i=3;
Main_NM_i=4;
Leis_WA_i=5;
Subs_WA_i=6;
Scho_WA_i=7;
Scho_NM_i=8;
Other_NM_i=9;

figure;
Ji=time*Static_i+1; %indexes 1-144
Jf=time*(Static_i+1);
subplot(10,1,1);
imagesc(V(Ji:Jf,eig_index)') %drawing the first components of the given index
title('Static','FontSize',11)
caxis([-0.10 0.10])
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

figure;
Ji=time*Subs_NM_i+1; %indexes 145-288
Jf=time*(Subs_NM_i+1);
subplot(10,1,2);
imagesc(V(Ji:Jf,eig_index)')
caxis([-0.10 0.10])
title('Subsistence Non-Motor','FontSize',11)
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {
'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

%figure;
Ji=time*Subs_CM_i+1; %indexes 289-432
Jf=time*(Subs_CM_i+1);
subplot(10,1,3);
imagesc(V(Ji:Jf,eig_index)')
caxis([-0.10 0.10])
title('Subsistence Cars & Motors','FontSize',11)
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {
'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

%figure;
Ji=time*Main_WA_i+1; %indexes 433-576
Jf=time*(Main_WA_i+1);
subplot(10,1,4);
imagesc(V(Ji:Jf,eig_index)')
caxis([-0.10 0.10])
title('Maintenance Walk','FontSize',11)
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {
'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

%figure;
Ji=time*Main_NM_i+1;
Jf=time*(Main_NM_i+1); %indexes 577-720
subplot(10,1,5);
imagesc(V(Ji:Jf,eig_index)')
caxis([-0.10 0.10])
title('Maintenance Non-motor','FontSize',11)
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {
'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

%figure;
Ji=time*Leis_WA_i+1; %indexes 721-864
Jf=time*(Leis_WA_i+1);
subplot(10,1,6);
imagesc(V(Ji:Jf,eig_index)')
caxis([-0.10 0.10])
title('Leisure Walk','FontSize',11)
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {
'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

%figure;
Ji=time*Subs_WA_i+1;
Jf=time*(Subs_WA_i+1); %indexes 864-1008
subplot(10,1,7);
imagesc(V(Ji:Jf,eig_index)')
title('Subsistence Walk','FontSize',11)
caxis([-0.10 0.10])
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

%figure;
Ji=time*Scho_WA_i+1;
Jf=time*(Scho_WA_i+1); %indexes 1009-1152
subplot(10,1,8);
imagesc(V(Ji:Jf,eig_index)')
title('School Walk','FontSize',11)
caxis([-0.10 0.10])
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

%figure;
colormap hot;
Ji=time*Scho_NM_i+1;
Jf=time*(Scho_NM_i+1); %indexes 1153-1296
subplot(10,1,9);
imagesc(V(Ji:Jf,eig_index)')
title('School Non-motor','FontSize',11)
caxis([-0.10 0.10])
set(gca,'XTick',0:24:144)
set(gca,'XTickLabel', {'00:00','04:00','08:00','12:00','16:00','20:00','24:00'})
set(gca,'YTick',[0.5 1.5])

%figure;
Ji=time*Other_NM_i+1;
Jf=time*(Other_NM_i+1); %indexes 1297-1440
subplot(10,1,10);
imagesc(V(Ji:Jf,eig_index)')
title('Others Non-motor','FontSize',11)
caxis([-0.10 0.10])
colorbar;
9.8 “Construction_Error.m”

% TPM reconstruction

function [errorRate] = Construction_Error(MBW,V,num_Eigen)

num = size(MBW,2);
V_Eigen = fliplr(V(:,num-num_Eigen+1:num));
Row_Eigen = V_Eigen';
samples=size(MBW,1);
Psi = mean(MBW,1);
A = zeros(size(MBW,1),size(MBW,2));

for i =1:samples
    A(i,:) = MBW(i,:) - Psi;
end

Construct_D = Row_Eigen*A';
W_Temp = V_Eigen*Construct_D;
W_Temp = W_Temp';
W = zeros(size(W_Temp,1),size(W_Temp,2));

for i =1:samples
    W(i,:) = W_Temp(i,:) + Psi;
end

% Get the reconstructed data
NewData = zeros(size(W,1),size(W,2));

for i=1:samples
    for t=1:144
        temp = [W(i,t), W(i,t+144), W(i,t+288), W(i,t+432), W(i,t+576), W(i,t+720), W(i,t+864), W(i,t+1008), W(i,t+1152), W(i,t+1296)];
        Mt = max(temp);
    end

    errorRate(i) = Mt;
for k=1:10
    if (W(i,t+144*(k-1))==Mt)
        NewData(i,t+144*(k-1))=1;
    end
end
end
end

% Check the reconstruction error
errorNum = 0;
for i=1:size(MBW,1)
    for j=1:size(MBW,2)
        if (NewData(i,j)~=MBW(i,j))
            errorNum = errorNum + 1;
        end
    end
end
end
errorRate = errorNum/(size(MBW,1)*size(MBW,2));

9.9 “Clustering.m”

% Clustering

% Kmeans cluster with PCA
Kmeans_D = Construct_D';
silh_Vector = zeros(10,1);
for i=2:10
    idx = kmeans(Kmeans_D,i);
    [silh,h] = silhouette(Kmeans_D,idx);
    silh_Vector(i) = mean(silh);
end

% Kmeans cluster with TPM matrix
% silh_Vector_TPM = zeros(10,1); % for i=2:10
%    idx_TPM = kmeans(TPM,i);
%    [silh_TPM,h_TPM] = silhouette(TPM,idx_TPM);
%    silh_Vector_TPM(i) = mean(silh_TPM); % end

% Diagnostics
figure
plot(linspace(2,10,9),silh_Vector(2:end),"--rs","LineWidth",2,'MarkerEdgeColor','k',
% hold on
plot(linspace(2,10,9),silh_Vector_TPM(2:end),'--bs','LineWidth',2,'MarkerEdgeColor','k','MarkerFaceColor','y','MarkerSize',8)
xlabel('Number of clusters','fontsize',11);
ylabel('Silhouette index','fontsize',11);
title('Kmeans clustering with PCA','fontsize',11);

% Choose 6 as the number of groups
idx6 = kmeans(Kmeans_D,6);

% Group 1
% Eigen-behavior
[C_G1] = create_C(MBW_G1);
[V_G1,D_G1] = eig(C_G1);
% eig_index_G1 = size(C_G1,1);
% draw_eigbehav(V_G1,eig_index_G1);
% draw_eigbehav(V_G1,eig_index_G1-1);
% draw_eigbehav(V_G1,eig_index_G1-2);
TPM_G1 = TPM((idx6==1),:);
MBW_G1 = MBW((idx6==1),:);
figure
colormap(jet(10))
imagesc(TPM_G1);
colorbar
set(gcf,'Colormap',mycmap);
set(gca,'XTick',0:12:144)
set(gca,'XTickLabel',
{'00:00','02:00','04:00','06:00','08:00','10:00','12:00','14:00','16:00','18:00','20:00','22:00','24:00'})
xlabel('Time of the Day','fontsize',11);
ylabel('Sample ID','fontsize',11);

% Group 2
TPM_G2 = TPM((idx6==2),:);
MBW_G2 = MBW((idx6==2),:);
figure
colormap(jet(10))
imagesc(TPM_G2);
colorbar
set(gcf,'Colormap',mycmap);
colorbar('YTickLabel', {'Static', 'Subs_NM', 'Subs_CM', 'Main_WA', 'Main_NM', 'Leis_WA', 'Subs_WA', 'Scho_WA', 'Scho_NM', 'Other_NM'});
set(gca,'XTick',0:12:144)
set(gca,'XTickLabel',
{'00:00', '02:00', '04:00', '06:00', '08:00', '10:00', '12:00', '14:00', '16:00', '18:00', '20:00', '22:00', '24:00'})
xlabel('Time of the Day','fontsize',11);
ylabel('Sample ID','fontsize',11);

% Group 3
TPM_G3 = TPM((idx6==3),:);
MBW_G3 = MBW((idx6==3),:);
figure
colormap(jet(10))
imagesc(TPM_G3);
colorbar
set(gcf,'Colormap',mycmap);
colorbar('YTickLabel', {'Static', 'Subs_NM', 'Subs_CM', 'Main_WA', 'Main_NM', 'Leis_WA', 'Subs_WA', 'Scho_WA', 'Scho_NM', 'Other_NM'});
set(gca,'XTick',0:12:144)
set(gca,'XTickLabel',
{'00:00', '02:00', '04:00', '06:00', '08:00', '10:00', '12:00', '14:00', '16:00', '18:00', '20:00', '22:00', '24:00'})
xlabel('Time of the Day','fontsize',11);
ylabel('Sample ID','fontsize',11);

% Group 4
TPM_G4 = TPM((idx6==4),:);
MBW_G4 = MBW((idx6==4),:);
figure
colormap(jet(10))
imagesc(TPM_G4);
colorbar
set(gcf,'Colormap',mycmap);
colorbar('YTickLabel', {'Static', 'Subs_NM', 'Subs_CM', 'Main_WA', 'Main_NM', 'Leis_WA', 'Subs_WA', 'Scho_WA', 'Scho_NM', 'Other_NM'});
set(gca,'XTick',0:12:144)
set(gca,'XTickLabel',
{'00:00', '02:00', '04:00', '06:00', '08:00', '10:00', '12:00', '14:00', '16:00', '18:00', '20:00', '22:00', '24:00'})
xlabel('Time of the Day','fontsize',11);
ylabel('Sample ID','fontsize',11);

% Group 5
TPM_G5 = TPM((idx6==5),:);
MBW_G5 = MBW((idx6==5),:);
figure
colormap(jet(10))
imagesc(TPM_G5);
colorbar
set(gca,'Colormap',mycmap);
colorbar('YTickLabel',
{['Static','Subs_NM','Subs_CM','Main_WA','Main_NM','Leis_WA','Subs_WA','Scho_WA','Scho_NM','Other_NM']);
set(gca,'XTick',0:12:144)
set(gca,'XTickLabel',
{'00:00','02:00','04:00','06:00','08:00','10:00','12:00','14:00','16:00','18:00','20:00','22:00','24:00'})
xlabel('Time of the Day','fontsize',11);
ylabel('Sample ID','fontsize',11);

% Group 6
TPM_G6 = TPM((idx6==6),:);
MBW_G6 = MBW((idx6==6),:);
figure
colormap(jet(10))
imagesc(TPM_G6);
colorbar
set(gca,'Colormap',mycmap);
colorbar('YTickLabel',
{['Static','Subs_NM','Subs_CM','Main_WA','Main_NM','Leis_WA','Subs_WA','Scho_WA','Scho_NM','Other_NM']);
set(gca,'XTick',0:12:144)
set(gca,'XTickLabel',
{'00:00','02:00','04:00','06:00','08:00','10:00','12:00','14:00','16:00','18:00','20:00','22:00','24:00'})
xlabel('Time of the Day','fontsize',11);
ylabel('Sample ID','fontsize',11);

% Trace back to Original ID so that we can calculate the demographics
ID_Group = zeros(size(ID_Trace,1),2);
ID_Trace = [ID_match_2,idx6];
for i=1:size(ID_Trace,1)
    ind = find(ID_match(:,2)==ID_Trace(i,1));
    ID_Group(i,1) = ID_match(ind,1);
    ID_Group(i,2) = ID_Trace(i,3);
end

% Here the matrix ID_Group contains two columns:
% First column - Original ID in the database
% Second column - Group index
% So now we can trace back to the database and get the demographics
% for each group

9.10 “Main.m”

%Main

%Color Matrix
figure
colormap(jet(10))
imagesc(TPM);
colorbar

% mycmap = get(gcf,'Colormap');
set(gcf,'Colormap',mycmap);
colorbar('YTickLabel',{'Static','Subs_NM','Subs_CM','Main_WA','Main_NM','Leis_WA','Subs_WA','Scho_WA','Scho_NM','Other_NM'});
set(gca,'XTick',[0:12:144])
set(gca,'XTickLabel','{00:00','02:00','04:00','06:00','08:00','10:00','12:00','14:00','16:00','18:00','20
:00','22:00','24:00'});
xlabel('Time of the Day','fontsize',11);
ylabel('Sample ID','fontsize',11);

% % Create colorbar
% axes=get(gcf,'CurrentAxes');
% colorbar('peer',axes,...
% % 'YTickLabel',{'Home','Work','School','Personal','Shopping','Leisure','Pick-up','Back-
% home','Back-to-office','Others'},...
% % 'ColorOrder',[0.752941191196442 0.752941191196442 0.752941191196442;0 0 1;1 1 0;0
% 0.749019622802734 0.749019622802734;1 0 0 1 0;0.800000000000000023841858 1;0
% 0 0;0.501960813999176 0 0]);

%Binary Matrix
MBW = generate_binary(TPM);
figure
colormap gray
imagesc(MBW)
colorbar
set(gca,'XTick',72:144:1440)
set(gca,'XTickLabel','{Static','Subs_NM','Subs_CM','Main_WA','Main_NM','Leis_WA','Subs_WA','Scho_WA','Sch
o_NM', 'Other_NM', 'fontsize', 11))
ylabel('Sample ID', 'fontsize', 11);

% Create matrix C
[C] = create_C(MBW);

% Eigen-behavior
[V, D] = eig(C);
eig_index = size(C, 1);
draw_eigbehav(V, eig_index);
draw_eigbehav(V, eig_index - 1);
draw_eigbehav(V, eig_index - 2);

% Plot ranked eigen-values
eigenValues = diag(D, 0);
num = length(eigenValues);
eigenValuesFirst50 = zeros(50, 1);
for i = 1:50
    eigenValuesFirst50(i) = eigenValues(num - i + 1);
end
figure
plot(eigenValuesFirst50, '--rs', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'MarkerSize', 4)
xlabel('Rank of eigen-vectors', 'fontsize', 13);
ylabel('Eigenvalues', 'fontsize', 13);

% Plot construction error
errorVector = zeros(50, 1);
for i = 1:50
    errorVector(i) = Construction_Error(MBW, V, i);
end
errorPercentageVector = errorVector * 100;
figure
plot(errorPercentageVector, '--rs', 'LineWidth', 2, 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'MarkerSize', 4)
xlabel('Number of eigen-vectors', 'fontsize', 13);
ylabel('Construction error(%)', 'fontsize', 13);

% Choose first 30 eigenvectors
num_Eigen = 30;
num = size(MBW, 2);
V_Eigen = fliplr(V(:, num-num_Eigen+1:num));
Row_Eigen = V_Eigen;
samples = size(MBW,1);
Psi = mean(MBW,1);
A = zeros(size(MBW,1),size(MBW,2));

for i = 1:samples
    A(i,:) = MBW(i,:) - Psi;
end

Construct_D = Row_Eigen*A';
W_Temp = V_Eigen*Construct_D;
W_Temp = W_Temp';
W = zeros(size(W_Temp,1),size(W_Temp,2));

for i = 1:samples
    W(i,:) = W_Temp(i,:) + Psi;
end

% Get the reconstructed data
NewData = zeros(size(W,1),size(W,2));

for i = 1:samples
    for t = 1:144
        temp = [W(i,t), W(i,t+144), W(i,t+288), W(i,t+432), W(i,t+576), W(i,t+720), W(i,t+864), W(i,t+1008), W(i,t+1152), W(i,t+1296)];
        Mt = max(temp);
        for k = 1:10
            if (W(i,t+144*(k-1)) == Mt)
                NewData(i,t+144*(k-1)) = 1;
            end
        end
    end
end

% Check the binary matrix with reconstructed matrix
figure
colormap gray
imagesc(NewData)
colorbar
set(gca,'XTick',72:144:1440)
set(gca,'XTickLabel',
    {'Static','Subs_NM','Subs_CM','Main_WA','Main_NM','Leis_WA','Subs_WA','Scho_WA','Sch o_NM','Other_NM','fontsize',11})
ylabel('Sample ID','fontsize',11);