CARNEGIE MELLON UNIVERSITY

DATA ANALYSIS PROJECT REPORT

Generating Activity Schedules for Human Agents Used in Simulations

Author:
Yu LIU

DAP Committee:
Professor Bill EDDY
Professor Ann LEE

*A paper submitted in fulfillment of the requirements
for the degree of Master of Science*

in the

Machine Learning Department
School of Computer Science

November 28, 2018

Contents

Chapter 1

Introduction

Agent-based models (ABM), which study autonomous agents' interactions in a constrained environment, are commonly used in epidemiology studies. By simulating simultaneous operations and interactions of multiple agents with varying characteristics, ABMs aim to reproduce manifestations of complex phenomena such as the spread of infectious diseases [1, 4, 6]. Such models rely on accurate and rich representations of the true population of interest as inputs in order to yield realistic results directly applicable to real life situations. Ideally, input populations should be representative of the true populations in terms of their geographic, demographic, behavioral and procedural characteristics. However, due to the great extent of comprehensiveness of the information required and limitation of physical resources, data of the entire true populations are almost always unavailable. Therefore, synthetic populations need to be learned and mimicked from the limited data drawn from the true populations.

The Models of Infectious Disease Agent Study (MIDAS) is a national research network that studies the spread of infectious diseases and develops response strategies through computational models. As a part of this network, the MIDAS research group at Carnegie Mellon University (CMU) Statistics Department led by Professor Bill Eddy has been working on building a synthetic ecosystem named Synthetic Populations and Ecosystems of the World (SPEW) [3], which may then serve as an input to ABMs. Supplying simulated populations generated using iterative proportional fitting, SPEW provides human agents with various demographic and geographic characteristics matching those from the true population, which are available from the U.S. census data. In comparison, however, data on the behavioral and procedural aspects of human populations are much more limited. Enriching human synthetic ecosystems with detailed, time-dependent information about individuals' activities will lead to more realistic ABMs and ultimately better-informed decision making. Therefore, for my Data Analysis Project, I focused on learning and simulating the behavioral and procedural aspects of the American population. Specifically, I worked on modeling individuals' daily activity schedules and incorporating synthetic agents' schedules into SPEW for use in more realistic, time-dependent ABMs.

The primary data source I used for studying human agents' activities is the American Time Use Survey, which consists of interviews of Americans' activity schedules on randomly selected days. The survey provides nationally-representative estimates of the American population's time use based on demographic and geographical features [7]. I explored both parametric and non-parametric simulation methods for generating activity schedules and evaluated those methods by comparing the key summary statistics of the generated activity sequences with those from the true surveys.

Chapter 2

Data

2.1 ATUS Overview

I used the American Time Use Survey (ATUS) published by the United States Bureau of Labor Statistics as the primary data source. ATUS serves to represent how, where, and with whom Americans spend their time and is the only federal survey collating data including non-market activities, such as socializing, childcare and volunteering. The survey samples households that have completed month 8 of the Current Population Survey (CPS) and randomly selects one respondent (age 15 and over) from each household. Respondents are asked only one time about how they spent their time on the previous day, where they were, and whom they were with through computer-assisted telephone interviews (CAT) [7]. ATUS data are collected on an ongoing, monthly basis. The survey results cover the time span of 15 years, from 2003 to 2017, with more than 190,000 interviews in total.

2.2 Activity Series

Each interview was recorded as a list of activity types and transition times. A daily schedule starts at 4:00 am and ends at 4:00 am the following day. Activity start and stop times are recorded at a precision of one minute. Activity types are categorized in a hierarchical manner, with three tiers in total, representing three levels of granularity. For example, personal care is considered a top-tier category, which can be divided into second-tier categories such as sleeping and grooming; sleeping, a second tier activity, can be further fine-grained into third-tier types such as actual sleeping and sleeplessness. There are in total 18 first-tier, 128 second-tier and over 400 third-tier categories. I adopted the first-tier categories in my simulation of activity types. If future needs for finer activity categorization arise, the proposed methods can be directly re-applied. The 18 categories are: Personal Care Activities, Household Activities, Caring for & Helping Household Members, Caring for & Helping Nonhousehold Members, Work & Work-Related Activities, Education, Consumer Purchases, Professional & Personal Care Services, Household Services, Government Services & Civic Obligations, Eating and Drinking, Socializing & Relaxing & Leisure, Sports & Exercise & Recreation, Religious and Spiritual Activities, Volunteer Activities, Telephone Calls, Traveling, Other/Unknown.

2.3 More Data on Predictive Features

Besides detailed activity logs, ATUS also includes linked data files including CPS responses, providing more background information about the respondents and their households, such as their employment status, occupation, age, gender, ethnicity and household membership. Specific predictive features of interviewees and their source data files are listed below:

Interview ID can be found in all data files (***TUCASEID***). This is the identifier used to link a respondent's interview records with all relevant demographic and geographic information.

Respondent's Age can be found in the Roster File (***TEAGE***). Before 2004, TEAGE was topcoded at 80, with all ages at and above 80 being recorded as 80. From 2005 afterwards, TEAGE was topcoded at 85. Despite some minor noise (less than 0.5% of all the interviews) as a result of this change in coding, I would still treat TEAGE as a continuous variable.

Respondent's Sex can be found in the Roster File (***TESEX***). TESEX takes a value between two categories - Male and Female.

Respondent's Weekly Income can be found in the Respondent File (***TRERNWA***). TRERNWA is the most commonly used income variable in ATUS. It is top-coded at $2884.61.

Top-coding Indicator for Respondent's Weekly Income can be found in the Respondent File (***TTWK***). TTWK is an indicator showing if TRERNWA is top-coded.

Respondent's Household Annual Income can be found in the CPS File (***HUFAMINC***, ***HEFAMINC***). Before 2010, the combined income of all family members during the last 12 months were recorded under HUFAMINC. From 2010 onwards, the same data were recorded under HEFAMINC. The household income I would use for prediction was taken by combining the two columns and filling in missing data (7%) by sampling from the valid income entries.

Number of Children ($<$ 18) in Respondent's Household can be found in the Respondent File (***TRCHILDNUM***). TRCHILDNUM takes the value between 0 and 30 (inclusive).

Number of Members in Respondent's Household can be found in the CPS File (***HRNUMHOU***). The number of household members might change between the CPS interview and the activity interview. It is the best estimate of the household size at the time of the interview.

State of Respondent's Household can be found in the CPS File (***GESTFIPS***). GESTFIPS is the FIPS code of the state in which the respondent's household is located.

Activity Series Recorded for a Day can be found in the Activity File (***TUACTIVITY_N***, ***TRTIER1P***, ***TUSTARTTIM***, ***TUSTOPTIME***). An activity chain is derived by combining all activity entries belonging to each respondent. The activity chain contains a list of activities happening in a 24 hour period (starting from 4 am), with each activity's type, start and stop times.

Chapter 3

Goal & Problem Representation

For the purpose of studying the spread of infectious diseases, knowing synthetic agents' activity locations and durations would be crucial in modeling their interaction and exposure to potential disease pathogens using agent-based models. A well simulated activity schedule should ideally capture both the time and space information about an individual's activities in a day. For this goal, I explored the following adaptations:

3.1 Activity Types

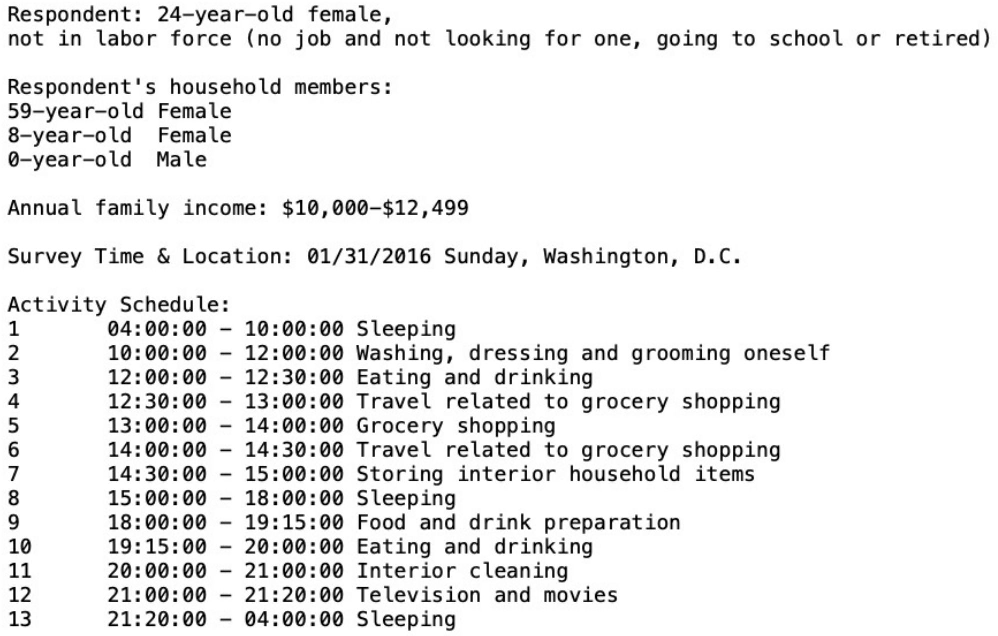
Activity types could be designed to directly capture the location information, since different activities are usually tied to distinct locations. I revisited the categorization of activities, looking into finer categorizations and trying to make sure activities grouped under the same category take place in similar locations. However, such a design turned out to be unfeasible as many activities, even down to the finest level of granularity, would not happen in homogeneous locations. For example, education can be done either at school (a public place) or at home (a private place), which are significantly different in the context of epidemic studies. Therefore, rather than assigning a single location to each activity type, we could instead record the location distribution for each activity type, and sample from the distribution if location information is specifically required in the future. This makes sure that simulated information about activity types is easily transferable to location knowledge.

3.2 Activity Durations

Together with activity types, activity durations directly define an activity chain for a day. I would set the time spent on each activity in a day as the key target in the simulation. Specifically, each activity chain (simulated or surveyed) can be summarized as a vector of total number of minutes spent on each type of activity (in this case, each vector has 18 elements corresponding to each of the 18 categories). The objective is thus to minimize the difference in the time-by-category vectors between a subject's simulated and true activity chains.

3.3 Prediction Objective

Therefore, our task is to generate a chain of activities defined by their types and durations (from which start and stop times can be directly derived) for each synthetic human agent with distinct characteristics. Below is an example drawn from the ATUS data - the data files include basic information about a respondent and his/her household, as well as a complete 24-hour activity schedule on a randomly selected day:



```
Respondent: 24-year-old female,
not in labor force (no job and not looking for one, going to school or retired)

Respondent's household members:
59-year-old Female
8-year-old  Female
0-year-old  Male

Annual family income: $10,000-$12,499

Survey Time & Location: 01/31/2016 Sunday, Washington, D.C.

Activity Schedule:
1       04:00:00 - 10:00:00 Sleeping
2       10:00:00 - 12:00:00 Washing, dressing and grooming oneself
3       12:00:00 - 12:30:00 Eating and drinking
4       12:30:00 - 13:00:00 Travel related to grocery shopping
5       13:00:00 - 14:00:00 Grocery shopping
6       14:00:00 - 14:30:00 Travel related to grocery shopping
7       14:30:00 - 15:00:00 Storing interior household items
8       15:00:00 - 18:00:00 Sleeping
9       18:00:00 - 19:15:00 Food and drink preparation
10      19:15:00 - 20:00:00 Eating and drinking
11      20:00:00 - 21:00:00 Interior cleaning
12      21:00:00 - 21:20:00 Television and movies
13      21:20:00 - 04:00:00 Sleeping
```

The survey data provide interviews for less than 200,000 individuals and our goal is to generate such activity schedules for any human agent with specific demographic and socio-economic characteristics from any synthetic population, which may easily contain over 300 million individuals.

Chapter 4

Methods

4.1 Methods I - Predicting Each Activity in a Schedule

One approach to produce new activity chains is to learn statistical properties of the schedules and generate each individual activity to form a chain, using probabilistic models.

An activity schedule consists of a series of activities, each defined by its activity type, start and end times. Since all day schedules start at 4:00 am (as in the ATUS interviews), knowing the duration of each activity would be sufficient to deduce its start and stop times. Therefore, predicting a schedule can be broken down into predicting the type and the duration of each activity in the chain.

4.1.1 Sampling Methods for Predicting Each Type and Duration

The goal of activity simulation is to generate synthetic activity sequences that are representative of the true distribution of various activity patterns exhibited by the true population. One way to perform such simulation is to build probabilistic models of the true population's activity patterns and sample from these models.

Markov Models for Predicting Types

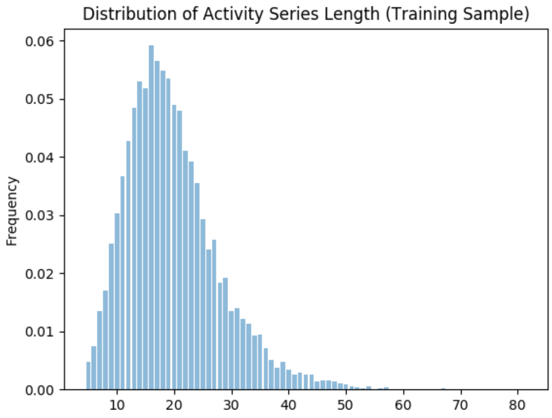
We can treat an individual agent's daily activities as different states in a discrete-time Markov chain and model the transition between activities by estimating the transition probabilities between different Markov states. With a transition probability matrix, we can then generate a random sequence of daily activities for each individual.

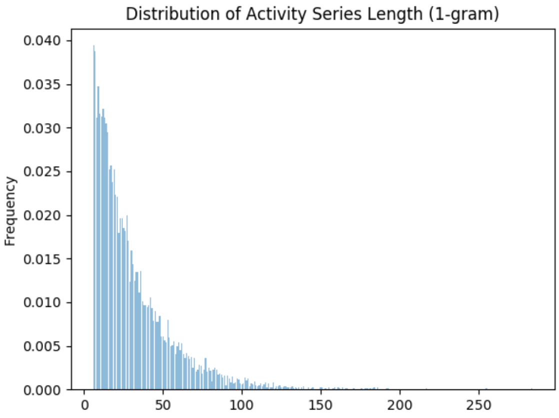
A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. By choosing the discrete-time Markov chain model, we make the following assumption about an individual's daily sequence of activities (denoted as a list of A_t's, where t refers to the t^{th} activity of the day):

$$P(A_t = a_t | A_{t-1} = a_{t-1}, ..., A_1 = a_1) = P(A_t = a_t | A_{t-1} = a_{t-1})$$

In other words, given an observed sequence of activities, the probability of a new activity happening next only depends on its previous activity. Given m possible types of activities, we need to estimate $\frac{m(m-1)}{2} = O(m^2)$ such transition probabilities. We can first use the maximum likelihood estimates, obtained by simply calculating the sample mean of each Bernoulli distribution representing a transition from state A_t to state A_{t+1}. To model the start and end probabilities, we can insert explicit start-of-day and end-of-day states to the observed activity sequences.

Based on my implementation, it seems that the maximum likelihood estimates are fairly consistent across different years, with small variances calculated using both sample estimates and the plug-in MLE method with a Bernoulli model. By adding an explicit start state and end state, we can generate an activity sequence even without the time factor - simply beginning with the start-of-day state and repeatedly sample the next activity from the transition probability table, until reaching the end-of-day terminate state. However, the length (i.e. the number of activities) of the generated sequences of activities turns out to have a significantly different distribution from that of the true (sample) distribution. As shown below, the simulated sequence lengths are highly skewed, with both a very long right tail, indicating the presence of excessively long sequences, and a very early peak, indicating a tendency for activity chains to end prematurely. Clearly, relying on Markov transitions alone to determine when to terminate an activity chain is not sufficient. It would be necessary to incorporate time constraints in our prediction.





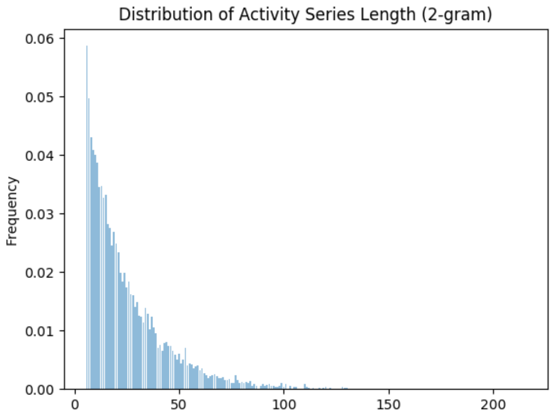
Markov Chain with N-grams

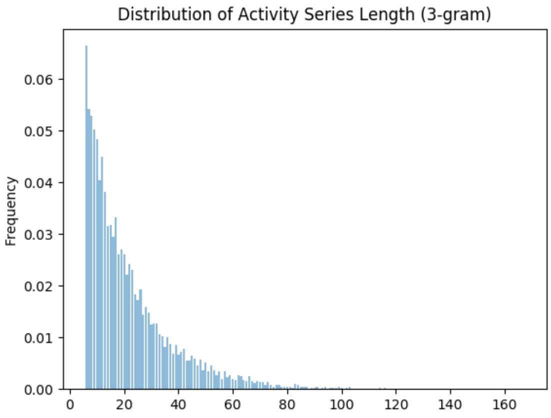
A further inspection of the activity sequences generated from the simple Markov model shows that many generated activity sequences end very quickly. A large portion of such sequences contain only three activities, with a "sleep - grooming (such as using the bathroom) - sleep" pattern. This is a very likely result from our simple transition model, as the transition probabilities between sleep and grooming are both high.

To prevent such cycling patterns, we propose to extend the one-gram model to n-gram models, calculating the transition probabilities based on previous n states rather than just one state, i.e.

$$P(A_t = a_t | A_{t-1} = a_{t-1}, ..., A_1 = a_1) = P(A_t = a_t | A_{t-1} = a_{t-1}, ..., A_{t-n} = a_{t-n})$$

Conditioning on the previous two activities being sleep and grooming, the probability of the current activity being sleep should be significantly reduced. Such a model would likely generate more realistic activity sequences. The total number of transition probabilities in a n-gram model would be $O(m^{n+1})$.





I modeled both 2-gram and 3-gram Markov transitions and the simulated activity sequences still appear to have similar skewed shapes as the simple 1-gram model. Nonetheless, I do observe that the sequence length distribution from the 2-gram model has a significantly shorter right tail than that of 1-gram, indicating a promising progress towards a more realistic model. However, as we further increase the memory window to 3-gram, the simulated sequence length distribution barely changes. Therefore, while the activity sequences seem to be closer to a Markov n-gram model, modeling transitions only is certainly not enough to guarantee an accurate simulation. We need to include more attributes, especially time factors, in our future models. Nonetheless, this shows that keeping a longer-memory model does result in closer-to-truth prediction results. It would be a good idea to try out both 1-gram and 2-gram models in any new Markov-based methods.

Geographical differences

I conducted a two-sample test on transition probabilities between different states in the U.S. and found that such probabilities differ significantly across states. For example, people living in Wyoming have a significantly higher probability of helping their household members after education activities as compared to those living in New York. Another example would be that Wyoming residents are significantly more likely to engage in sports activities after socializing as compared to those in Florida.

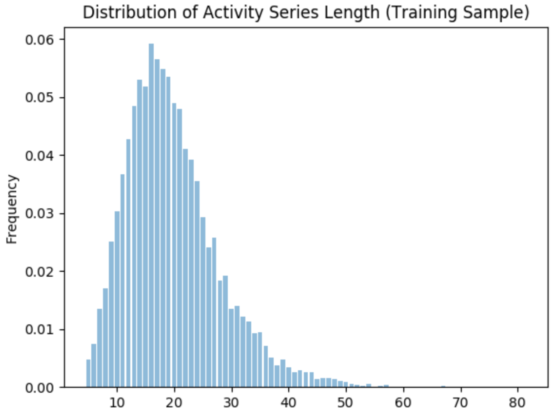
Indeed, activity patterns differ across geographical regions. Geo-location should thus be incorporated as an important conditioning feature in future models. I will thus use a separate model for each state.

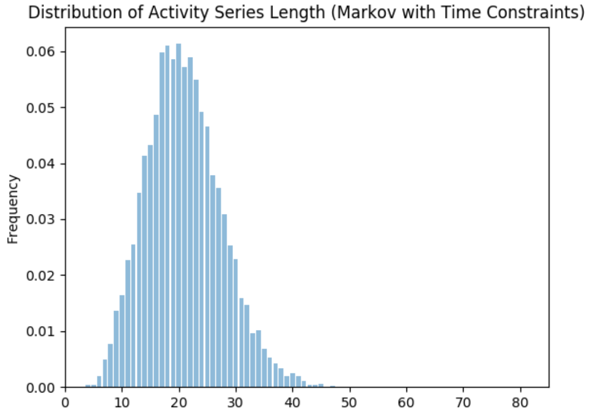
Markov Chain with Duration Density Estimates

From running the simple transition models, we have observed simulated activity sequences with lengths varying from 3 (very simple "sleep - grooming - sleep" pattern) to over 100. Considering an individual has on average roughly 12-16 hours of daily activity time excluding sleep, filling these hours with either one single activity or 60 activities would be unpractical. I thus decided to introduce time constraints in our model to address this problem.

To begin, we need to estimate the duration of each simulated activity. I have run histogram density estimation for activity duration for each activity type. To generate activity sequences, We first sample a new activity based on the transition probability matrix as described earlier and then sample an activity duration from its corresponding duration histogram, subtracting the used time from the total available time of the day before continuing to choose the next activity. We end a sequence upon finishing the daily time quota (24 hours).

As expected, this method effectively controls the length of the generated activity chains. As shown below, the simulated chain length distribution is much less skewed and largely resembles the true length distribution.


Distribution of Activity Series Length (Training Sample)

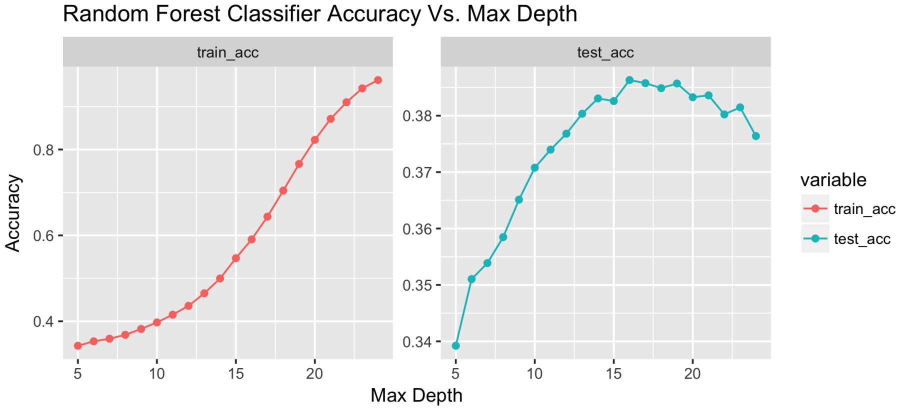

Distribution of Activity Series Length (Markov with Time Constraints)

4.1.2 Conditional Methods for Predicting Each Type and Duration

The previous sampling methods relies on probabilistic activity models directly learned from the true population's activity surveys. However, such learned models are also population-specific, and typically do not apply well to a new population. For example, we observe that activity transition probabilities vary significantly across geo-locations. This is not surprising as we would naturally expect people from different places and with different demographic characteristics and socio-economic backgrounds have different activity patterns. For higher generalization power, it is important to learn to predict activities based on individuals' demographic and socio-economic features, optimizing predictions for individuals rather than an entire population. Therefore, we will explore conditional models, such as classification and regression models, in learning activity patterns.

Activity Type Classification

Previously using Markov, we observe that remembering more previous states improves our simulation, and the model should naturally be the best when we remember all history states prior to the current step. However, remembering a long history would be expensive - in terms of both memory space and time complexity. Instead, we can use a summarized version of the history to predict for each new event - storing the number of occurrences and the total time spent on each type of activity, and use these counts and durations as input features (together with demographic features and time of a day) to a random forest classification model. Using an ensemble of 100 decision tree classifiers with a maximum depth of 17, the classifier achieves around 38% test prediction accuracy, which is still reasonable, given the volatility of human behavior.



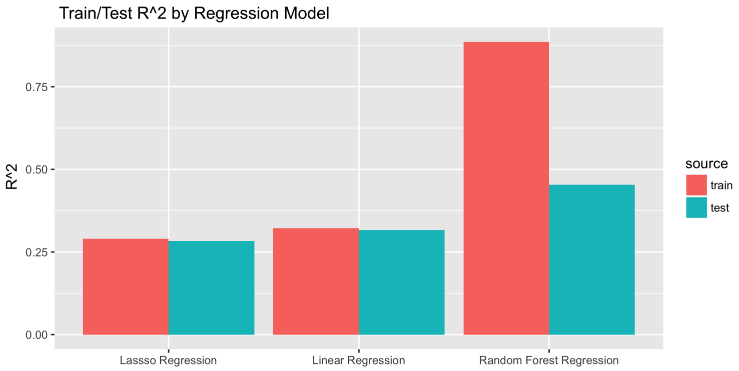
Activity Duration Regression

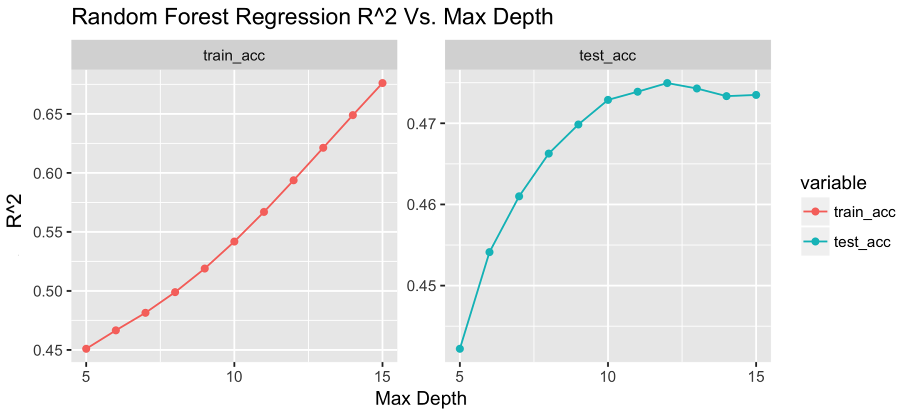
Similarly, a regression model can be used to predict each simulated activities' duration, conditioning on not only the current activity type, but also the individual's demographic traits, previous activity counts and durations, as well as the time of the day.

First performing a simple linear regression with coefficients significance t-test (shown on the next page), I found out the current activity type (**TRTIER1P#**) is a strongly significant predictor for activity durations, which fits our expectation. A majority of previous activity counts (**COUNT_#**) are significant, and so are some of the previous duration counts (**DUR_#**). The individual's age (**TEAGE**), sex (**TESEX**), household income (**HFAMINC**) and the number of household children (**TRCHILDDNUM**) also affect activity durations in a significant manner. I will keep these predictors for the regression model.

```
Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept)  1.712e+02  2.216e+00   77.248  < 2e-16 ***
TRTIER1P2   -1.071e+02  1.018e+00 -105.229  < 2e-16 ***
TRTIER1P3   -1.151e+02  1.466e+00  -78.530  < 2e-16 ***
TRTIER1P4   -1.120e+02  2.710e+00  -41.317  < 2e-16 ***
TRTIER1P5    3.307e+01  1.542e+00   21.448  < 2e-16 ***
TRTIER1P6   -1.332e+01  4.030e+00   -3.307 0.000945 ***
TRTIER1P7   -1.117e+02  1.707e+00  -65.417  < 2e-16 ***
TRTIER1P8   -1.061e+02  4.139e+00  -25.642  < 2e-16 ***
TRTIER1P9   -1.100e+02  9.023e+00  -12.194  < 2e-16 ***
TRTIER1P10  -1.076e+02  1.630e+01   -6.603 4.07e-11 ***
TRTIER1P11  -1.173e+02  1.140e+00 -102.849  < 2e-16 ***
TRTIER1P12  -5.976e+01  9.952e-01  -60.046  < 2e-16 ***
TRTIER1P13  -7.227e+01  2.743e+00  -26.350  < 2e-16 ***
TRTIER1P14  -7.684e+01  3.232e+00  -23.772  < 2e-16 ***
TRTIER1P15  -6.651e+01  3.837e+00  -17.333  < 2e-16 ***
TRTIER1P16  -1.200e+02  2.741e+00  -43.767  < 2e-16 ***
TRTIER1P18  -1.320e+02  9.636e-01 -136.971  < 2e-16 ***
TRTIER1P50  -8.979e+01  2.864e+00  -31.354  < 2e-16 ***
TUSTARTTIM   2.277e-02  1.366e-03   16.666  < 2e-16 ***
COUNT_1     -1.087e+00  3.324e-01   -3.269 0.001078 ** 
COUNT_2     -2.468e+00  1.909e-01  -12.925  < 2e-16 ***
COUNT_3     -6.866e-01  2.469e-01   -2.781 0.005413 ** 
COUNT_4     -2.761e+00  5.529e-01   -4.993 5.94e-07 ***
COUNT_5     -4.674e+00  4.463e-01  -10.472  < 2e-16 ***
COUNT_6     -3.002e+00  1.492e+00   -2.011 0.044311 *  
COUNT_7     -2.103e+00  5.440e-01   -3.866 0.000111 ***
COUNT_8     -7.925e-01  1.249e+00   -0.634 0.525812    
COUNT_9     -6.081e+00  2.284e+00   -2.662 0.007770 ** 
COUNT_10    -3.502e+00  5.088e+00   -0.688 0.491264    
COUNT_11     5.547e-01  5.111e-01    1.085 0.277807    
COUNT_12    -2.990e+00  3.032e-01   -9.864  < 2e-16 ***
COUNT_13    -5.181e+00  9.586e-01   -5.405 6.51e-08 ***
COUNT_14    -2.971e+00  1.152e+00   -2.579 0.009901 ** 
COUNT_15    -1.309e+00  1.164e+00   -1.125 0.260714    
COUNT_16    -4.460e+00  9.723e-01   -4.588 4.49e-06 ***
COUNT_18    -3.539e-01  2.208e-01   -1.603 0.108932    
COUNT_50    -1.219e+00  8.652e-01   -1.409 0.158984    
DUR_1       -5.488e-02  2.916e-03  -18.824  < 2e-16 ***
DUR_2        2.289e-02  4.083e-03    5.607 2.07e-08 ***
DUR_3        6.846e-03  7.686e-03    0.891 0.373116    
DUR_4        8.386e-03  1.272e-02    0.659 0.509578    
DUR_5       -1.001e-02  3.092e-03   -3.239 0.001200 ** 
DUR_6       -2.087e-02  1.061e-02   -1.967 0.049162 *  
DUR_7        1.576e-02  1.029e-02    1.531 0.125708    
DUR_8        4.597e-03  2.355e-02    0.195 0.845278    
DUR_9        7.394e-02  5.183e-02    1.427 0.153696    
DUR_10      -1.785e-03  9.276e-02   -0.019 0.984652    
DUR_11       6.429e-02  1.174e-02    5.474 4.40e-08 ***
DUR_12       7.250e-02  3.370e-03   21.517  < 2e-16 ***
DUR_13       1.258e-02  1.185e-02    1.062 0.288189    
DUR_14       1.211e-02  1.575e-02    0.769 0.442007    
DUR_15      -8.901e-03  1.439e-02   -0.619 0.536230    
DUR_16       7.968e-02  3.146e-02    2.533 0.011310 *  
DUR_18       5.075e-02  7.181e-03    7.068 1.58e-12 ***
DUR_50       2.671e-02  1.356e-02    1.970 0.048851 *  
TEAGE       -1.714e-01  2.054e-02   -8.345  < 2e-16 ***
TRERNWA      2.872e-04  5.128e-04    0.560 0.575521    
TRCHILDNUM  -2.271e+00  4.955e-01   -4.582 4.60e-06 ***
HFAMINC     -5.507e-01  8.444e-02   -6.522 6.99e-11 ***
HRNUMHOU     4.256e-01  3.762e-01    1.131 0.257893    
TTWK        -3.984e+00  2.760e+00   -1.443 0.148918    
TESEX       -3.910e+00  6.234e-01   -6.273 3.56e-10 ***
```

Using an ensemble of 100 decision tree regressors with a maximum depth of 12, random forest regression can achieve an R^2 (coefficient of determination) of around 47%, which is also a reasonable score, implying that the model has accounted for almost 50% of variability in human's activity time.





4.2 Method II - Predicting a Schedule

Modeling sequences of activities with various types and different duration (spanning from 1 minute to 10 hours) is challenging, due to the high dimensional search space. Considering the complex nature of our prediction data type, it can be difficult to find adequate conditional models given the limited amount of data available. Alternatively, we can directly assign entire schedules from the surveys to new individuals. This can be done by first mapping individuals to a feature-vector space, identifying similar survey subjects and assigning their recorded activity schedules to be predictions for new individuals. With minimal model constraints, we will be able to preserve more complex dependency relations in activity sequences, thereby generating much more realistic activity chains.

4.2.1 Nearest Neighbors Methods

We can use the nearest-neighbor approach to match a synthetic individual with a survey subject based on their demographic characteristics. Kristian Lum [5] proposed a similar method for sampling activity schedules: for each target synthetic agent, we may identify his/her nearest neighbor by first finding the best-matching household to his/her own household, and then select the best-matching individual from the matched household. We can then use the matched subject's activity schedule record as a new simulated chain for the target agent. In the context of ATUS data, only one respondent from each selected household is interviewed. Therefore, we will simply incorporate household information as part of a subject's demographic features and directly compute person-to-person distances to find the nearest neighbor.

The key to nearest neighbor methods is to select a proper distance metrics. Assuming the true population demographics (denoted by X) and the synthetic demographics (denoted by X^*) have the same distribution, a few distance metrics I will explore for computing survey subject i (P_i) and synthetic agent i^* ($P^*_{i^*}$)'s dissimilarity include:

Euclidean Distance Using Demographic Covariates

$$Dist(P_i, P^*_{i^*}) = ||X_i - X^*_{i^*}||$$

This is a naive distance metric that simply aggregates differences in every dimension (i.e. in every demographic feature). To prevent the distance from being dominated by features with large variances, input feature standardization is necessary.

Mahalanobis Distance Using Demographic Covariates

$$Dist(P_i, P^*_{i^*}) = (X_i - X^*_{i^*})^T S^{-1} (X_i - X^*_{i^*})$$

Relaxing Euclidean distance's strong assumption about the data being isotropically Gaussian, we may choose Mahalanobis distance, which models anisotropical Gaussian data. However, as Lum pointed out, the computation weighs covariates simply based on their variances and covariances, without taking into account their true effects on the activity series output [5]. For example, if covariances between all pairs are 0, all predictors are weighted equally, which may not be ideal - the Mahalanobis distance can be significantly affected by many irrelevant predictors while being affected minimally by a few highly relevant predictors.

Mahalanobis Distance Using Fitted Values

$$Dist(P_i, P^*_{i^*}) = (\hat{Y}_i - \hat{Y}^*_{i^*})^T S^{-1} (\hat{Y}_i - \hat{Y}^*_{i^*})$$

Ideally, we would like to adopt a distance scheme which automatically assigns greater weights to predictors with more bearings on the final activity schedules. Lum proposed running a simple conditional model on summary statistics of the output sequences, such as a linear regression $Y_k = f_k(X) + \epsilon$ for estimating total time spent on each activity k (i.e. a summary statistic of the final activity schedule), and then use the fitted and predicted values to compute the Mahalanobis distance instead [5]. Using a simple regression model can help us easily rule out irrelevant features, as their estimated coefficients will be close to zero, lending them little weight in calculating the fitted values and therefore the final distances.

4.2.2 Decision Tree Approach

Straightforward as the nearest neighbor approach is, its effectiveness highly depends on the correct choice of distance metrics, which requires knowledge about each predictive feature and their relative importance in predicting the final activity chain. Decision tree, on the other hand, is an alternative non-parametric model that simply learns decision rules to divide data examples into subgroups without an explicitly stated distance metric. Neither does it require extensive data pre-processing such as normalization or pre-assigning feature importance. With less assumptions, decision trees can potentially find decision rules that generalize better and even learn feature importance through training. In addition, decision tree models should generally perform faster in prediction time, as decision paths traversing down a tree should be in $O(log\ n)$, as compared to $O(n)$ for the nearest-neighbor approach, with n being the number of training data points. I will try out two decision tree methods to match survey and synthetic individuals.

Classification Tree

We can reduce this into a classification problem, with each class corresponding to a distinct training example (and with a distinct activity chain). I construct a decision tree with the CART algorithm, using Gini impurity as the splitting criterion. In an un-regularized version, such an algorithm produces a tree with all pure leaves - each leaf containing a single training example (or identical ones, which are unlikely given the high dimensionality of the input features in this case). Effectively, such a tree contain decision rules that quickly distinguish training examples from one another (each occupying a rectangular sub-space in the decision space). For each new example, we can find a decision path down the tree into a leaf χ_j, and we will then assign the same-leaf training example's activity chain to be the new example's prediction. This is similar to finding a nearest neighbor for a new example, except that the "distance" measure is more flexible, varying for different combinations of input feature values. As Chen suggested [2], such decision trees implicitly learn similarity relations between individuals and are hence adaptive nearest neighbor methods. The ID of the best-matching subject P_i for a synthetic individual $P^*_{i^*}$ is thus found by the decision tree rule $T(P^*_{i^*})$ defined as follows:

$$T(P^*_{i^*}) = \max_i \sum_{j=1}^{J} 1\{P^*_{i^*} \in \chi_j \wedge P_i \in \chi_j\}$$

$$Schedule(ID_{P^*_{i^*}}) = Schedule(ID_{T(P^*_{i^*})})$$

Regression Tree

Since we aim to minimize the difference in the total time spent on each activity between our prediction and the truth, another way to construct the decision tree is to directly minimize such differences in the leaves. We can construct a regression tree instead, using the time-by-activity vectors as the target variables and apply CART to minimize the mean square error of the vectors within each leaf. With this new objective, the decision rules may be tuned towards grouping individuals with similar activity patterns together, thus leading to more realistic simulation results. The decision rule for finding the best-matching subject is then identical to the classification approach.

Addressing Overfitting

Unregularized trees are easily prone to overfitting problems. Imposing Occam's Razor principle to promote simpler tree structures would likely increase the generalization power of our final models. I proceeded to experiment with constraining the minimum number of examples per leaf, using cross-validation to determine the best degree of tree regularization. With more than one training examples in a leaf, predictions are made by randomly sampling one such example's activity chains for a new synthetic agent assigned to the same leaf.

The problem of overfitting may also be addressed by using an ensemble method, i.e. using random forest classifier/regressor. Using ensemble models, predictions can be made by taking the majority vote of the most likely best-matching training example from all decision tree estimators l:

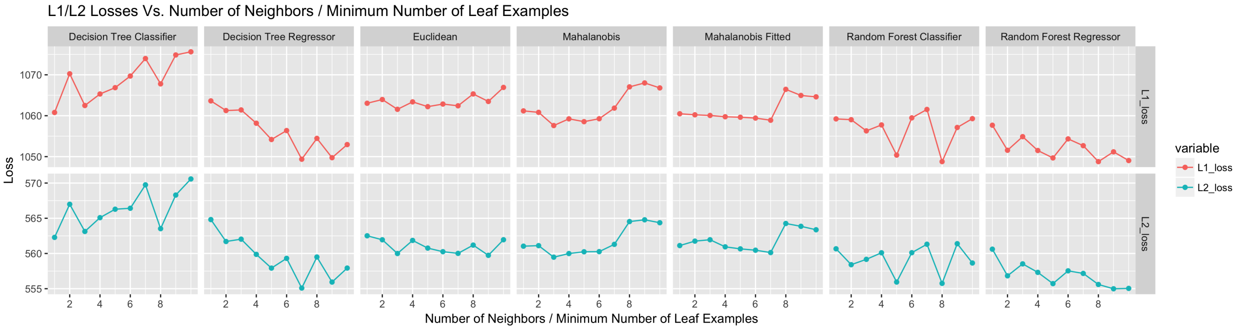
$$T(P^*_{i^*}) = \max_i \sum_{l=1}^{L} \sum_{j=1}^{J^l} 1\{P^*_{i^*} \in \chi^l_j \wedge P_i \in \chi^l_j\}$$

$$Schedule(ID_{P^*_{i^*}}) = Schedule(ID_{T(P^*_{i^*})})$$

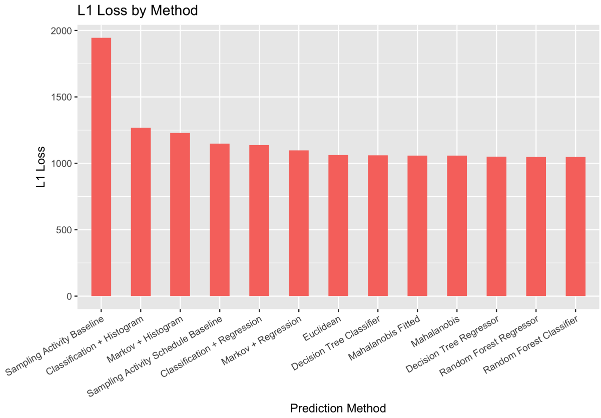
Chapter 5

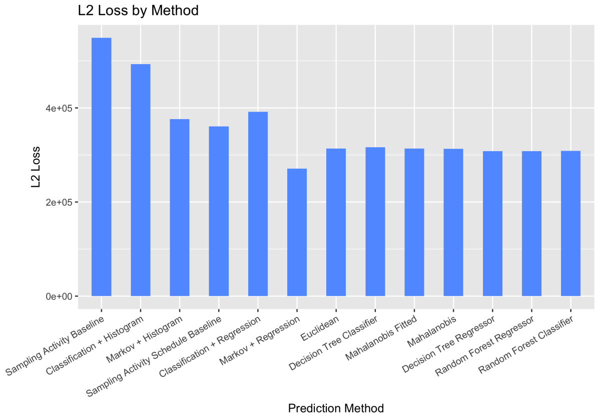
Results

As discussed, we decide to use the total time spent on each activity as the summary statistics to assess the quality of simulated activity sequences. For each proposed method, we will first compare the L1 and L2 distances between the total time spent on each activity from predicted sequences and the true sequences. Before comparing performances between different methods, we first compare the performances with different degrees of smoothing - i.e. different number of neighbors to consider in the case of nearest neighbor sampling and different minimum number of leaf examples required in the case of decision tree sampling.



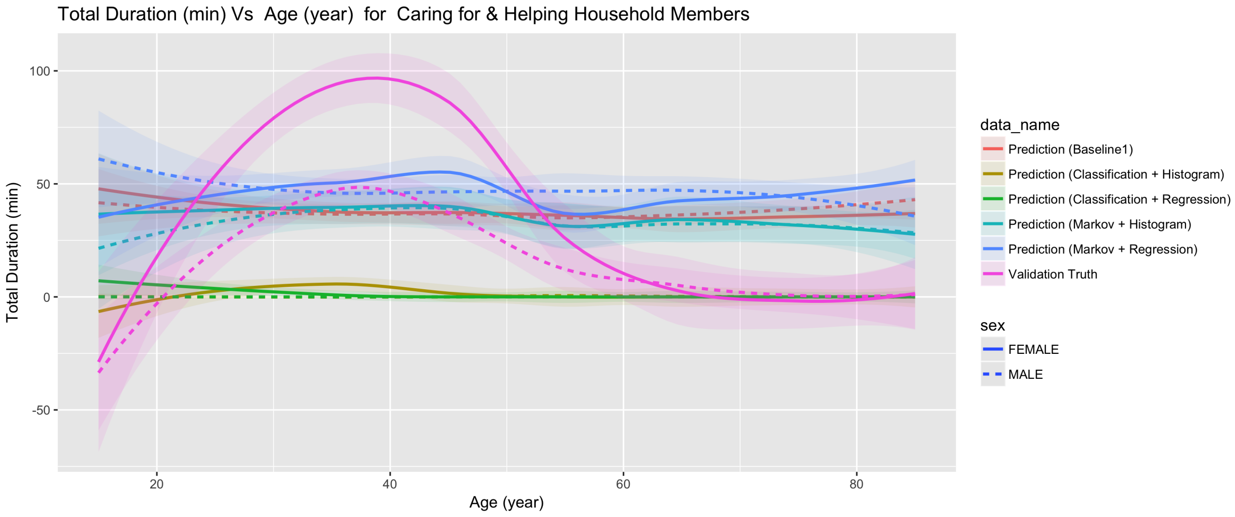
The L1 and L2 distance metrics gave fairly consistent results. Different methods clearly have different optimal degree of smoothing. We will pick the smoothing factor (k) that minimizes both L1 and L2 losses for each algorithm and proceed to compare these methods against each other:





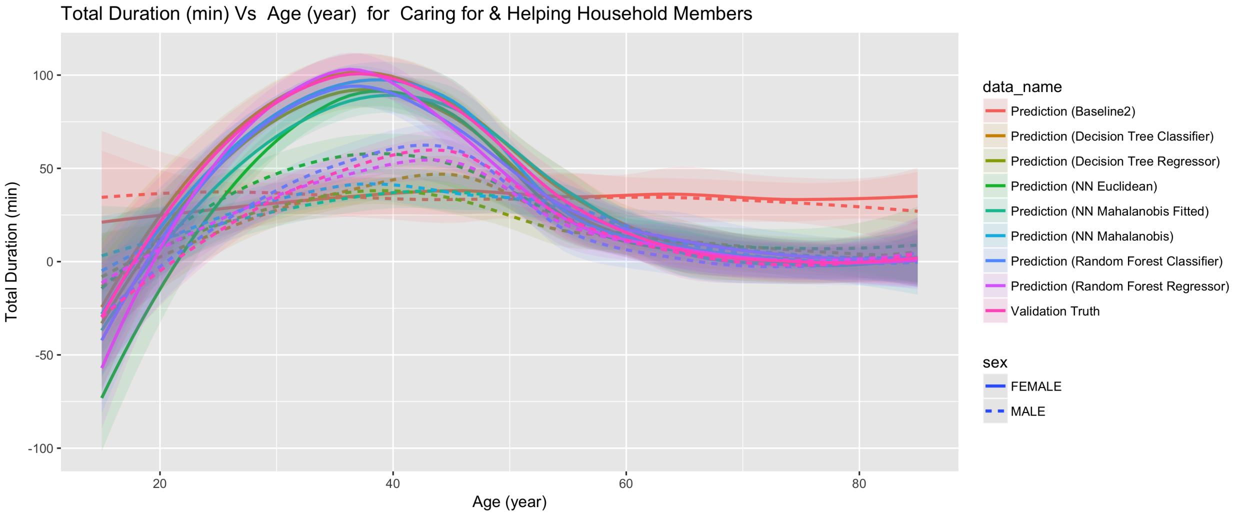
All sampling methods achieve significant improvement from the baseline model (random sampling). Despite their relatively close performances, the decision tree regressor and decision tree ensemble methods - the random forest classifier and regressor - seem to achieve the lowest errors overall. Decision tree methods generally work better than nearest neighbor sampling approaches, as the former make less assumptions about predictors' importance without explicitly assuming a fixed distance metric. Also, ensemble methods are clearly effective in preventing the overfitting problem commonly found in single decision trees and thus improving generalization power. Moreover, the decision tree regressor, which was targeted to optimize the time-vector difference directly, is already much stronger than a decision tree classifier, which was built for a surrogate optimization goal (distinction between examples based on demographic features).

We will also examine how total time spent on each type of activity varies with important demographic factors such as age and gender. This is done by plotting smoothed regression lines of the predictions and compare them against the results from the true ATUS sample. An ideal simulation method should produce curves with similar shapes with those from the true sample. Below is an example of the curves of total time spent on helping household members, disaggregated by age and gender.



The pink lines (solid and dotted) show the curves for the total time spent by male and female across all ages on helping household members. While both lines peak in the age of middle 30's, the peak is significantly sharper and higher for the female than the male; the difference between the two curves diminishes as the total time spent on helping household membmers decline with age.

The first baseline model (**Baseline1**, i.e. generating individual activities by randomly sampling activity type and using mean activity duration) clearly fails to capture any trend - the red lines corresponding to the baseline method are both flat at a level around 40 minutes. Similarly for other methods, there is no obvious gap between the dotted line and the solid line of each color, indicating that the first group of methods cannot effectively capture the gender difference between total activity time patterns. Moreover, most smoothed regression lines from predictions are flat, retaining little information about how activity total durations vary with age. Evidently, despite their fairly good performance in L1/L2 losses (especially the Markov + Regression method), methods which generate individual activities to form schedules are inadequate for modeling the complex activity patterns exhibited by individuals with distinct characteristics.



In the above plot for the second group of methods, we can see clearly from the plotted orange lines that the predictions from the baseline model (**Baseline2**, i.e. selecting existing activity schedules by randomly sampling survey responses) contain no information about the true trend either - the two orange lines are both flat and close to each other. Curves from other sampling methods capture the trend to various degrees. In this case, predictions from the random forest classifier and regressor (the dark blue and violet lines respectively) seem to follow the truth curves most closely. Clearly, the ensemble decision tree methods give the most accurate predictions, which are effectively very close to the truth.

Chapter 6

Conclusion

Agent-based models (ABM) are popular tools used to study autonomous agents' interactions and hence complex phenomena such as the spread of infectious diseases. ABMs require accurate and rich representations of the true population of interest as inputs in order to produce realistic results. The SPEW system developed by the CMU MIDAS group serves to provide such a synthetic input that is demographically and geographically representative of the true populations of the world. This project aims to incorporate detailed, time-dependent information about individual agents' activities into SPEW. Specifically, we would like to learn from limited interview records from the ATUS data to generate for each synthetic individual a comprehensive activity sequence, consisting of a list of activities defined by their types and durations.

There are in general two approaches to generate new activity chains: one way is to learn statistical properties about activity patterns in a chain and use probabilistic models to predict individual activities to form a chain; the other way is to directly assign existing activity schedules from survey records to synthetic agents using specific candidate-matching schemes. For the first group of methods, I started by modeling activity type transitions using markov chain models and sampling activity durations from duration histograms. To improve the generalization power to other populations with different demographic and socio-economic compositions, I proceeded to adopt conditional methods to predict each new activity type and duration based on individuals' demographic characteristics and previous activities in the chain. For the second group of activities, I experimented with different candidate selection schemes such as nearest neighbor with different distance metrics and CART with different optimization criteria.

For evaluation, I targeted the summary statistic of the complex activity chains - the total time (in minutes) spent on each type of activity by individuals. I compared the performance of the proposed methods with two baseline methods - generating each new activity by randomly sampling its type and using the corresponding type's mean duration (first group approach) as well as randomly sampling an existing activity schedule to assign to a new individual (second group approach). Comparing the L1/L2 losses on the total time by activity types, all proposed methods achieve significant improvement from the first baseline model, but only second-group methods and one first-group method beat the second baseline significantly. Upon examining the trend of total time spent per activity against demographics such as age and gender, second-group methods prove to be much stronger in preserving the true trend. Specifically, I found the candidate selection method using a random forest classifier has superior performance in all evaluations. Therefore, I will adopt this method to generate for synthetic human agents in SPEW.

Bibliography

[1] Ajelli, Marco, et al. "Comparing large-scale computational approaches to epidemic modeling: agent-based versus structured metapopulation models." BMC infectious diseases 10.1 (2010): 190.

[2] Chen, George H., and Devavrat Shah. "Explaining the Success of Nearest Neighbor Methods in Prediction." Foundations and Trends® in Machine Learning 10.5-6 (2018): 337-588.

[3] Gallagher, Shannon, et al. "SPEW: synthetic populations and ecosystems of the world." Journal of Computational and Graphical Statistics just-accepted (2018): 1-30.

[4] Kumar, Supriya, et al. "Policies to reduce influenza in the workplace: impact assessments using an agent-based model." American journal of public health 103.8 (2013): 1406-1411.

[5] Lum, Kristian, et al. "A two-stage, fitted values approach to activity matching." International Journal of Transportation 4.1 (2016).

[6] Perez, Liliana, and Suzana Dragicevic. "An agent-based approach for modeling dynamics of contagious disease spread." International journal of health geographics 8.1 (2009): 50.

[7] "ATUS News Releases." U.S. Bureau of Labor Statistics, U.S. Bureau of Labor Statistics, www.bls.gov/tus/.