PERSONa

A Synthetic Mobility Dataset Generator for Next-Place Prediction

Università della Svizzera Italiana

Caselli Alex

June 2020

1 PERSONA synthetic dataset

This dataset born from the need of mobility traces provided with demographics data of the users.

PERSONA has three main demographic groups (or main classes): (1) the student class; (2) the employee class and (3) the retired class.

Each main class has six possible relevant places and three subclasses which are specializations of the main class. Additionally, an extra generic place *'other'* is provided to all the classes to represent any other possible relevant place which could not be identified or known.

Each class specialization has a subset of the relevant places of the parent class. This allows different subclasses to have similar but not identical relevant places, augmenting the variance and the realism of the generated mobility traces.

		Home	School	Park	University	Library	Station	Work	Parents' House	Grocery Store	Gym	Restaurant	Church
Student	Type A	X	X	X	-	-	-	-	-	-	-	-	-
	Type B	X	-	X	X	X	-	-	-	-	-	-	-
	Type C	X	-	-	X	-	X	-	-	-	-	-	-
Employee	Type A	X	-	-	-	-	-	X	X	X	-	-	-
	Type B	X	-	-	-	-	X	X	-	-	X	-	-
	Type C	X	-	-	-	-	-	-	X	-	X	-	-
Retired	Type A	X	-	X	-	-	-	-	-	X	-	X	-
	Type B	X	-	-	-	-	-	-	-	X	-	X	X
	Type C	X	-	X	-	-	-	-	-	-	X	X	-

Tabella 1: Main classes, sub-classes and their relevant places.

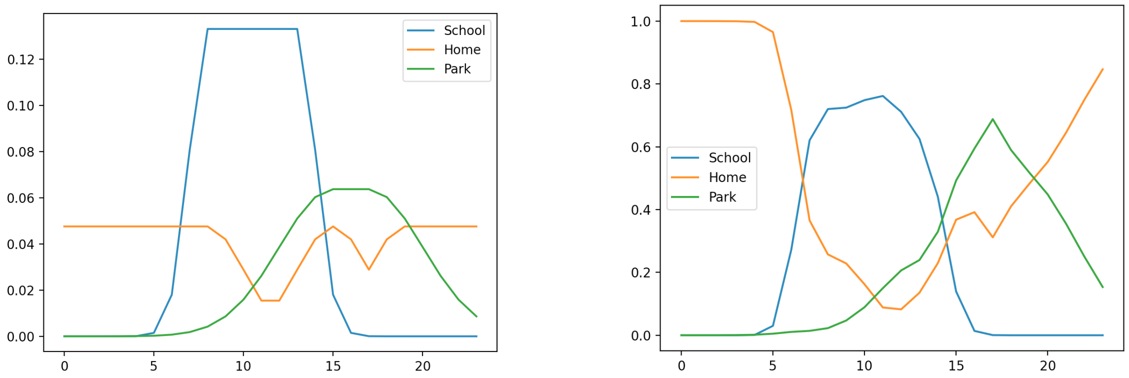
An example can be given by the student class, while master's and high school's students fit both in the student category, their movements and relevant places can very easily differ. While is more likely for a master's student to come to its university from an another city by train, a teenager in the high schools probably can reach its school by feet or by short-range public transports like bike-sharing or buses. Each subclass is defined simply as a Python dictionary. For this reason, new classes and subclasses can be defined very easily by adding or editing elements in the dictionary. To generate mobility traces, each place should have, similarly to what states have in Markov Models, a starting probability and transition probabilities from and towards any place for each hour of the day; Possibly with differences between week and weekend days.
The proper way to represent these probabilities is with matrices where each subclass have a matrix with starting place probabilities which describes for each relevant places of the subclass, the probability to start the day in that place. Additionally, place transition probabilities would require three-dimensional matrices to be described taking into account the fact that these probabilities would depend not only on the current place but also on the time of the day. While the starting place can be easily fixed to a default location such as *Home*; Overall this system would require the definition, considering an average of 4 places for each one of the 9 subclasses, of (((4 places x 4 places x 24 hours) x 9 users values for the transition probabilities matrices. For a total number of 3,456 values to decide by hand. This is clearly infeasible. In order to overcome this definition problem,

PERSONA uses an approximation of the probabilities matrices.
The solution is simple but effective, instead of defining probabilities for each hour of the day, different Gaussians ditributions are fitted in the main hours of each place. Each place has hours in which is more likely for a user to be, as for a student is more likely to be at school at 10 while is very improbable for him to be there at 23. Each subclass has so a set of hours in which is highly probable to be in a specific place, this set of hours are used as means for several normal distributions. While the mean specifics the hot-hours for that location, the standard deviation specifies instead how well defined that hours are and how much those hot-hours can vary.

The different Gaussians defined for each users' subclass and each place are then combined to obtain a unique probability value for each hour of the day. To merge overlapping probabilities three methods have been considered: (1) take the maximum probability, (2) take the minimum probability or (3) take the average probability. From different tests, taking the maximum provided resulting distributions closer to the desired one.
These probabilities can describe only one place at time, to generate movements between different places transition probabilities are required. This requires an additional function, normalize. The scope of normalizing is to make the sum of all the probabilities of different places at the same hour time slot equal to the desired probability value P.

Because probability distributions of places are defined individually, they could present different scales and the sum of the probabilities of different places can be either close to zero or larger than 1.0.

The normalization function operates on all the places for each time slot and solves the scale problem, an example of the result of the application of the function is given by figure 1.



(a) Probability distributions pre-normalization

(b) Probability distributions post-normalization

Figura 1: Comparison between the probability distributions of three different place before and after normalization.

Probabilities are stored into a two-dimensional array where for each hour of the day there is a list with a probability value for each place in that time slot. Since each entry of the generated mobility trace is contained in a time of fixed length (one hour in this case), the mobility trace is called *'Slotted'*.

Definition 1.1. Slotted Mobility Trace. We define the mobility trace Ω_s of a user u related to a time interval Δ_t lists the ordered sequence of places visited by the user in that time interval by considering time windows of length s. It can be defined as: $\Omega_s(u, \Delta_t) = \left\langle p_{(0,s]}, p_{(s,2s]}, ..., p_{(t*s,(t+1)*s]} \right\rangle$

The location of the user at a given time t is then simply random sampled from the correspondent list of place probabilities. To further replicate realistic human mobility, a stability value is used to increase the probability of the "*stay-in-place*" action, making more likely for a user to stay in the current place instead of change location frequently.

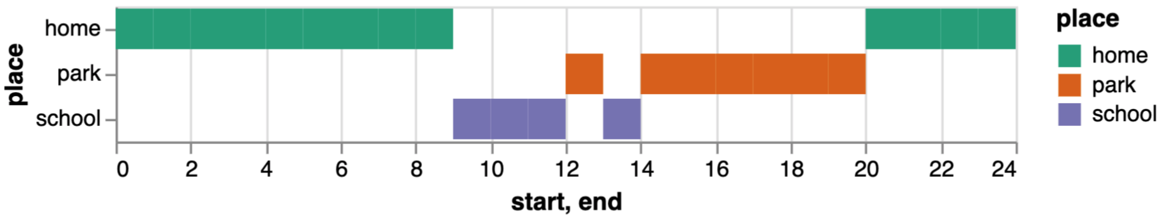


Figura 2: Simulation of a synthetic slotted mobility trace produced by a student.

Very often, routines of people change between week and weekend days. Students may not go to school or to the university in the weekends, and so the probability of visit those places should be very low in those days of the week. To implement this distinction, each place's dictionary has an additional field which specifies if the location is visited only during the week, only during the weekend or in both cases. Whether a weekday is simulated, a place which usually is visited only during the weekends gets as normalization threshold a maximum probability of 0.1 that makes it very unlikely to be sampled.

The generation of the mobility traces is simple, each user's subclass has the same probability to be sampled.

N subclasses of users are selected randomly. For each day of the D days that needs to be generated, 24 places (one per hour) are sampled from the probability distributions for each of the N users. Each record is stored together with other information such as the userID; The user's class; The time and the previous location, into a pandas dataframe which has the schema illustrated in table 2.

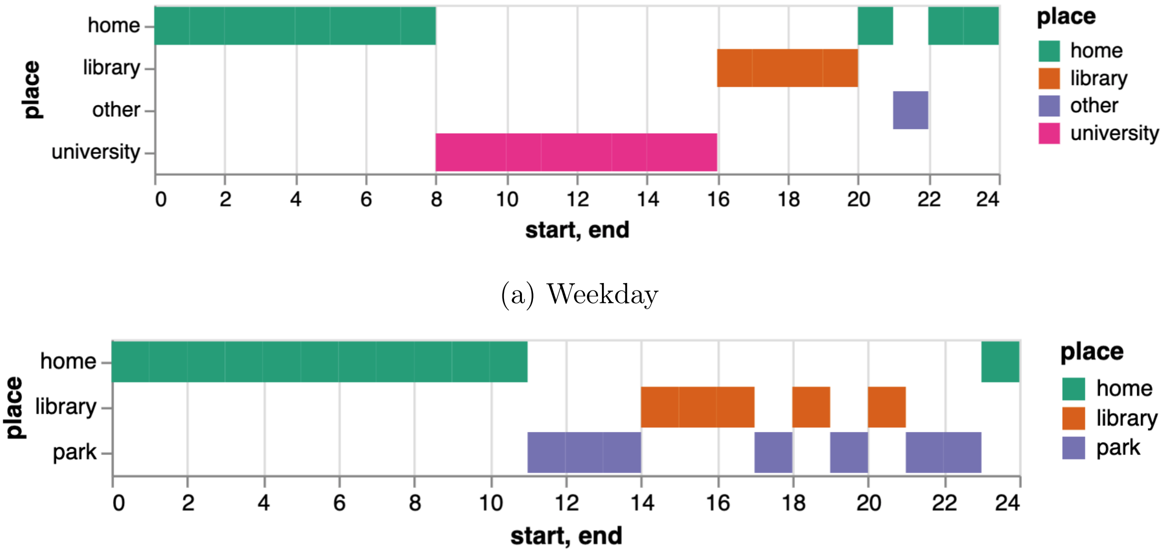
Algorithm 1 PERSONA dataset generation

```
function GENERATE_TRACES($D, N$)    ▷ Generated D days of traces for N users
    Generate N users by randomly select between the defined subclasses
    $dataframe = DataFrame()$
    for $day = 0, 1, 2, \ldots, D - 1$ do
        for $user = 0, 1, 2, \ldots, N - 1$ do
            $p_{hour} \leftarrow' Home'$    ▷ Start from Home
            $trajectory \leftarrow [p_{hour}]$
            for $hour = 1, 2, \ldots, 23$ do
                $p_{hour} \leftarrow sampleFromPlaces(user, day, hour, p_{hour})$
                $trajectory.append(p_{hour})$    ▷ Add the sampled place to the trajectory
            end for
            $dataframe.add(trajectory)$
        end for
    end for
    return $dataframe$
end function
```

Name	Description	Values range
users_id	Unique user's id	[0, N-1]
users_class	Main class type of the user	[0, 2]
start_hour	Hour of the day when the user is at start_place	[0, 23]
end_hour	Hour of the day when the user is at end_place	[0, 23]
start_place	Initial relevant place of the user	[0, 12]
end_place	Final relevant place of the user	[0, 12]
day_type	Whether is a weekday or a weekend	[0, 1]

Tabella 2: Final pre-processed PERSONA dataset schema.

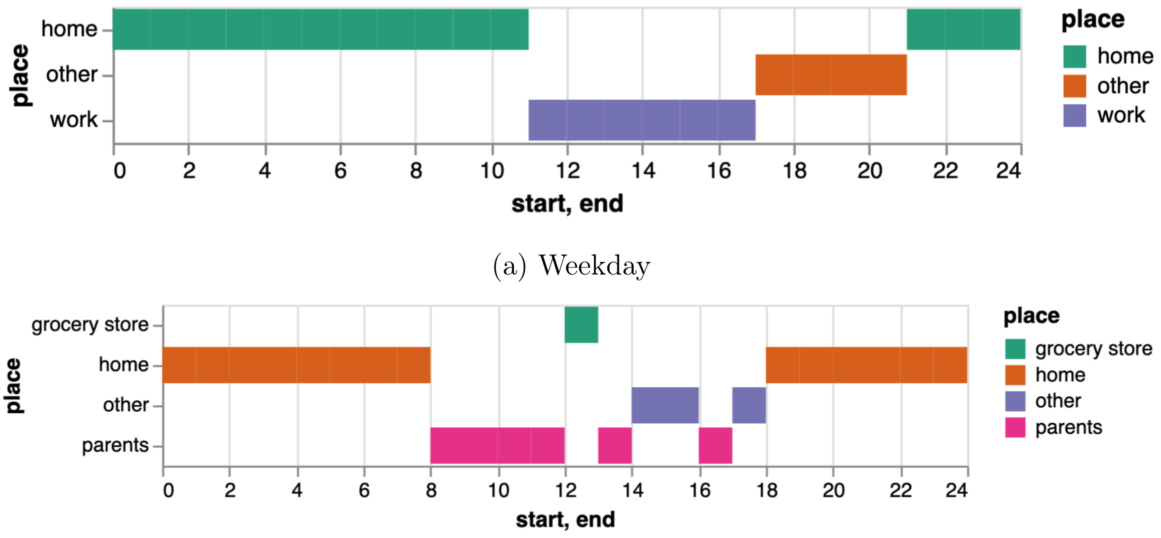
Figures 3, 4 and 5 provide an example for both week and weekend days for each user class generated.



(a) Weekday

(b) Weekend

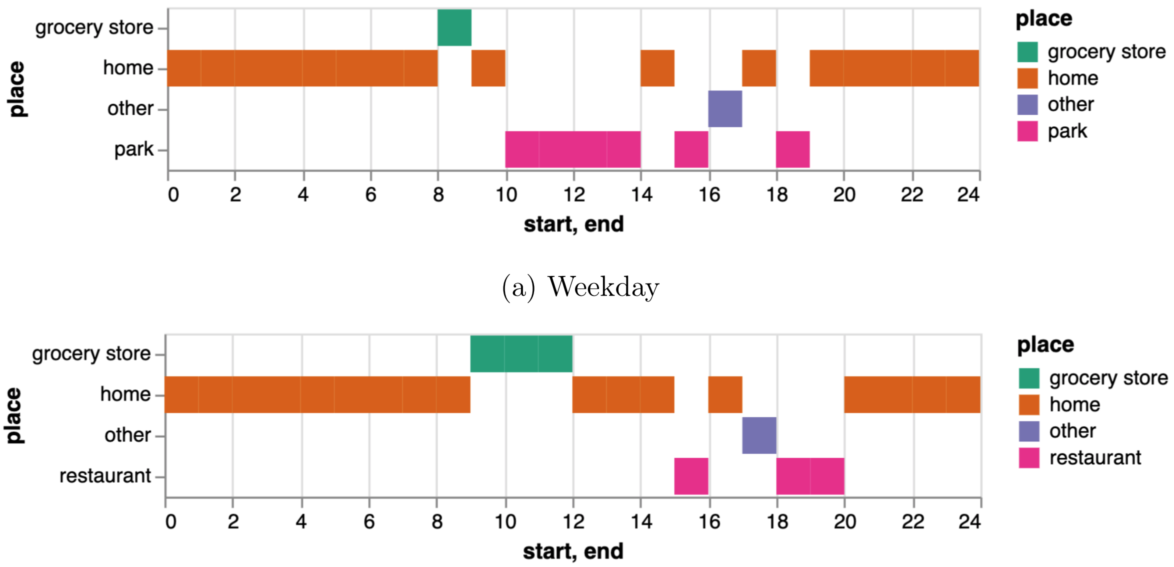
Figura 3: Mobility traces generated for an university's student in a week and a weekend day.



(a) Weekday

(b) Weekend

Figura 4: Mobility traces generated for an employee in a week and a weekend day.



(a) Weekday

(b) Weekend

Figura 5: Mobility traces generated for a retired in a week and a weekend day.

The final dataset generated with PERSONA comprehends 30 days of records of 60 unique users.