Technical Report for 2.3.2 Mode Choice for Education Tour

Siyu LI

June 2016

Model Description

This model will predict the mode choice for education tour. Education tour can only be conducted by full time student (we have such constrains in day pattern model) and the destination has been given. It is only necessary to predict the mode.

Choice Set

There are 9 modes in total.

- Public bus: the mode is public bus.
- MRT/LRT: the mode is either MRT or LRT.
- Private bus: the mode is school bus/company bus/shuttle bus.
- Drive alone: the mode is car/van/lorry driver and only the driver is on board.
- Shared ride 2: the mode is car/van/lorry driver or passenger and 2 persons are on board.
- Shared ride 3+: the mode is car/van/lorry driver or passenger and 3 or more persons are on board.
- Motorcycle: the mode is either motorcycle rider or passenger.
- Walk: the mode is walk
- Taxi: the mode is taxi

The structure of choice set is nested.

Model Structure

The Day Pattern Model is a NL model.

```
# V1 = public bus
# V2 = MRT
# V3 = private bus
# V4 = drive alone
# V5 = shared2
# V6 = shared3+
# V7 = motor
# V8 = walk
# V9 = taxi

# the nest structure is:
#     car = MU1 , [4,5,6,7]
#     PT = MU2 , [1,2,3]
#     other = 1.0, [8,9]

V1 = cons_bus +
    beta1_1_tt * tt_bus_ivt +
    beta1_2_tt * tt_bus_walk +
    beta1_3_tt * tt_bus_wait +
    beta_cost * cost_bus +
    beta_central_bus * Central_dummy +
    beta_transfer * average_transfer_number +
    beta_female_bus * Female_dummy +
    age_over_15*beta_age_over_15_bus +
    university_student * beta_university_student_bus

V2 = cons_mrt +
    beta1_1_tt * tt_mrt_ivt +
    beta1_2_tt * tt_mrt_walk +
    beta1_3_tt * tt_mrt_wait +
    beta_cost * cost_mrt +
    beta_central_mrt * Central_dummy +
    beta_transfer  * average_transfer_number +
    beta_female_mrt * Female_dummy +
    age_over_15*beta_age_over_15_mrt +
    university_student * beta_university_student_mrt

V3 = cons_privatebus +
    beta_private_1_tt * tt_privatebus_ivt +
    beta_cost * cost_privatebus +
    beta_central_privatebus * Central_dummy +
    beta_distance*(d1+d2) +
    beta_residence * residential_size +
    beta_attraction*school_attraction +
    beta_residence_2*residential_size**2 +
    beta_attraction_2*school_attraction**2 +
    beta_female_privatebus* Female_dummy +
    age_over_15*beta_age_over_15_private_bus +
    university_student * beta_university_student_private_bus

V4 = cons_drive1 +
    beta2_tt_drive1 * tt_cardriver_all +
    beta_cost * cost_cardriver +
    beta_female_drive1 * Female_dummy +
    beta_zero_drive1 * zero_car +
    beta_oneplus_drive1 * one_plus_car +
    beta_twoplus_drive1 * two_plus_car +
    beta_threeplus_drive1 * three_plus_car +
    age_over_15*beta_age_over_15_drive1 +
    university_student * beta_university_student_drive1

V5 = cons_share2 +
    beta2_tt_share2 * tt_carpassenger_all +
    beta_cost * cost_carpassenger/2.0 +
    beta_central_share2 * Central_dummy +
    beta_female_share2 * Female_dummy +
    beta_zero_share2 * zero_car +
    beta_oneplus_share2 * one_plus_car +
    beta_twoplus_share2 * two_plus_car +
    beta_threeplus_share2 * three_plus_car +
    age_over_15*beta_age_over_15_share2 +
    university_student * beta_university_student_share2


V6 = cons_share3 +
    beta2_tt_share3 * tt_carpassenger_all +
    beta_cost * cost_carpassenger/3.0 +
    beta_central_share3 * Central_dummy +
    beta_female_share3 * Female_dummy +
    beta_zero_share3 * zero_car +
    beta_oneplus_share3 * one_plus_car +
    beta_twoplus_share3 * two_plus_car +
    beta_threeplus_share3 * three_plus_car +
    age_over_15*beta_age_over_15_share3 +
    university_student * beta_university_student_share3

V7 = cons_motor +
    beta2_tt_motor * tt_motor_all +
    beta_cost * cost_motor +
    beta_central_motor * Central_dummy +
    beta_zero_motor * zero_motor +
    beta_oneplus_motor * one_plus_motor +
    beta_twoplus_motor * two_plus_motor +
    beta_threeplus_motor * three_plus_motor +
    beta_female_motor * Female_dummy +
    age_over_15*beta_age_over_15_motor +
    university_student * beta_university_student_motor +
    beta_distance_motor * (d1+d2)

V8 = cons_walk +
    beta_tt_walk * tt_walk +
    beta_central_walk * Central_dummy +
    beta_female_walk * Female_dummy +
    age_over_15*beta_age_over_15_walk +
    university_student * beta_university_student_walk

V9 = cons_taxi +
    beta_tt_taxi * tt_taxi_all +
    beta_cost * cost_taxi +
    beta_central_taxi * Central_dummy +
    beta_female_taxi * Female_dummy +
    age_over_15*beta_age_over_15_taxi +
    university_student * beta_university_student_taxi


#Estimated values for all betas
#Notice: the betas that not estimated are fixed to zero.

cons_bus = Beta('bus cons',0,-10,10,0) = -1.94
cons_mrt = Beta('MRT cons',0,-10,10,0) = -2.93
cons_privatebus=Beta('private bus cons',0,-10,10,0) = -2.21
cons_drive1=Beta('drive alone cons',0,-10,10,1) = 0
cons_share2=Beta('share2 cons',0,-10,10,0) = -5.66
cons_share3=Beta('share3 plus cons',0,-10,10,0) = -5.44
cons_motor=Beta('motor cons',0,-20,10,0) = -6.50
cons_walk=Beta('walk cons',0,-10,10,0) = -0.386
cons_taxi=Beta('taxi cons',0,-10,10,0) = -6.12

beta1_1_tt = Beta('travel time beta1_1 ivt',0,-10,10,0) = -0.623
beta1_2_tt = Beta('travel time beta1_2 waiting',0,-10,10,0) = -0.703
beta1_3_tt = Beta('travel time beta1_3 walk',0,-10,10,0) = -1.09

beta_private_1_tt=Beta('travel time beta_private_1 ivt',0,-10,10,0) = -0.692
```

```
beta2_tt_drive1 = Beta('travel time beta drive1',0,-10,10,0) = -0.900
beta2_tt_share2 = Beta('travel time beta share2',0,-10,10,0) = -1.31
beta2_tt_share3 = Beta('travel time beta share3',0,-10,10,0) = 1.13
beta2_tt_motor = Beta('travel time beta motor',0,-10,10,1) = 0

beta_tt_walk =Beta('travel time beta walk',0,-10,10,0) = -3.52
beta_tt_taxi =Beta('travel time beta taxi',0,-10,10,0) = -2.54

beta_cost= Beta('generic cost beta',0,-15,15,0) = -0.0184
beta4_2_cost = Beta('travel cost beta4_2',0,-15,15,1) = 0
beta5_2_cost = Beta('travel cost beta5_2',0,-15,15,1) = 0
beta6_2_cost = Beta('travel cost beta6_2',0,-15,15,1) = 0
beta7_2_cost = Beta('travel cost beta7_2',0,-15,15,1) = 0
beta8_2_cost = Beta('travel cost beta8_2',0,-15,15,1) = 0
beta9_2_cost = Beta('travel cost beta9_2',0,-15,15,1) = 0
beta10_2_cost = Beta('travel cost beta10_2',0,-15,15,1) = 0


beta_central_bus=Beta('central dummy in bus',0,-10,10,0) = 0.214
beta_central_mrt=Beta('central dummy in mrt',0,-10,10,0) = 0.397
beta_central_privatebus=Beta('central dummy in privatebus',0,-10,10,0) = 0.753
beta_central_share2=Beta('central dummy in share2',0,-10,10,0) = 0.395
beta_central_share3=Beta('central dummy in share3 plus',0,-10,10,0) = 0.486
beta_central_motor=Beta('central dummy in motor',0,-10,10,0) = 0.00692
beta_central_taxi=Beta('central dummy in taxi',0,-10,10,0) = 0.959
beta_central_walk=Beta('central dummy in walk',0,-10,10,0) = 0.0671


beta_female_bus=Beta('female dummy in bus',0,-10,10,0) = 0.826
beta_female_mrt=Beta('female dummy in mrt',0,-10,10,0) = 0.948
beta_female_privatebus=Beta('female dummy in privatebus',0,-10,10,0) = 0.886

beta_female_drive1=Beta('female dummy in drive1',0,-10,10,1) = 0
beta_female_share2=Beta('female dummy in share2',0,-10,10,0) = 0.864
beta_female_share3=Beta('female dummy in share3 plus',0,-10,10,0) = 0.753

beta_female_motor=Beta('female dummy in motor',0,-10,10,1) = 0
beta_female_taxi=Beta('female dummy in taxi',0,-10,10,0) = 0.724
beta_female_walk=Beta('female dummy in walk',0,-10,10,0) = 0.953


beta_zero_drive1=Beta('zero cars in drive1',0,-10,10,1) = 0
beta_oneplus_drive1=Beta('one plus cars in drive1',0,-10,10,1) = 0
beta_twoplus_drive1=Beta('two plus cars in drive1',0,-10,10,1) = 0
beta_threeplus_drive1=Beta('three plus cars in drive1',0,-10,30,1) = 0
```

```
beta_zero_share2=Beta('zero cars in share2',0,-10,10,1) = 0
beta_oneplus_share2=Beta('one plus cars in share2',0,-10,10,0) = 2.73
beta_twoplus_share2=Beta('two plus cars in share2',0,-10,10,0) = 1.28
beta_threeplus_share2=Beta('three plus cars in share2',0,-10,10,0) = 0.121

beta_zero_share3=Beta('zero cars in share3 plus',0,-10,10,1) = 0
beta_oneplus_share3=Beta('one plus cars in share3 plus',0,-10,10,0) = 2.61
beta_twoplus_share3=Beta('two plus cars in share3 plus',0,-10,10,0) = 0.963
beta_threeplus_share3=Beta('three plus cars in share3 plus',0,-30,10,1) = 0


beta_zero_motor=Beta('zero motors in motor',0,-10,10,1) = 0
beta_oneplus_motor=Beta('one plus motors in motor',0,-10,10,1) = 0
beta_twoplus_motor=Beta('two plus motors in motor',0,-10,10,1) = 0
beta_threeplus_motor=Beta('three plus motors in motor',0,-10,10,1) = 0

beta_transfer=Beta('average transfer number in bus and mrt', 0,-10,10,1) = 0

beta_distance=Beta('distance in private bus',0,-10,10,0) = -0.00836

beta_residence=
Beta('home zone residential size in private bus',0,-10,10,0)= -0.466

beta_residence_2=
Beta('square of home zone residential size in private bus',0,-10,10,1) = 0

beta_attraction=
Beta('work zone work attraction in private bus',0,-10,10,0) = -0.0393

beta_attraction_2=
Beta('square of work zone work attraction in private bus',0,-10,10,1) = 0

MU1 = Beta('MU for car',1,1,100,0) = 1.45
MU2 = Beta('MU for PT', 1,1,100,0) = 1.51

beta_age_over_15_bus=
Beta('Beta for age over 15 dummy in bus',0,-10,10,0)= 2.00

beta_age_over_15_mrt=
Beta('Beta for age over 15 dummy in mrt',0,-10,10,0)= 2.54

beta_age_over_15_private_bus=
Beta('Beta for age over 15 dummy in private bus',0,-10,10,1) = 0

beta_age_over_15_drive1=
Beta('Beta for age over 15 dummy in drive1',0,-10,10,1) = 0

beta_age_over_15_share2=
Beta('Beta for age over 15 dummy in share2',0,-10,10,0) = 0.414

beta_age_over_15_share3=
Beta('Beta for age over 15 dummy in share3',0,-10,10,1) = 0

beta_age_over_15_motor=
Beta('Beta for age over 15 dummy in motor',0,-10,10,1) = 0

beta_age_over_15_walk=
Beta('Beta for age over 15 dummy in walk',0,-10,10,0) = 1.36

beta_age_over_15_taxi=
Beta('Beta for age over 15 dummy in taxi',0,-10,10,0) = 1.14

beta_university_student_bus=
Beta('Beta for university student dummy in bus',0,-10,10,0) = -0.184

beta_university_student_mrt=
Beta('Beta for university student dummy in mrt',0,-10,10,0) = 0.341

beta_university_student_private_bus=
Beta('Beta for university student dummy in private bus',0,-10,10,0) = 0.220

beta_university_student_drive1=
Beta('Beta for university student dummy in drive1',0,-10,10,1) = 0

beta_university_student_share2=
Beta('Beta for university student dummy in share2',0,-10,10,0) = 0.191

beta_university_student_share3=
Beta('Beta for university student dummy in share3',0,-10,10,1) = 0

beta_university_student_motor=
Beta('Beta for university student dummy in motor',0,-10,10,1) = 0

beta_university_student_walk=
Beta('Beta for university student dummy in walk',0,-10,10,1) = 0

beta_university_student_taxi=
Beta('Beta for university student dummy in taxi',0,-10,10,0) = 1.88

beta_distance_motor=Beta('distance in motor',0,-10,10,1) = 0
```

Variables

This section will introduce how to acquare variables needed in estimating the model. Variable name, describtion, accessing method or generating pseudo code will be provided. In the mode choice model, the tour origin is home and the MTZ for home can be acquired in `population.txt`. The tour destination is known to be the usual work location, whose MTZ can be also acquired from `population.txt`. In the following, we use `origin` and `destination` to represent MTZ of tour origin (home) and MTZ of tour destination (usual work location).

Previous to generating related variables, data sets (`am`,`pm`,`op` and `zone_employment`) can be stored in dicts for fast querying.

```
# From gen.py
import numpy as np
am=np.genfromtxt(file_dir+'AMcosts.dat',names=True)
pm=np.genfromtxt(file_dir+'PMcosts.dat',names=True)
op=np.genfromtxt(file_dir+'OPcosts.dat',names=True)
zone_employ=np.genfromtxt(file_dir+'zone_employment.txt',names=True)
AM={};PM={}; OP={}; ZONE={}

for row in am:
    AM[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['AM2dis'], \
    'car_ivt' : row['AM2Tim']/60,\
    'pub_ivt':  row['AM2ivt']/60, \
    'pub_out':  (row['AM2aux']+row['AM2wtt'])/60, \
    'pub_wtt': row['AM2wtt']/60,\
    'pub_walkt': row['AM2aux']/60,\
    'car_cost_erp': row['AM2ERP']/100, \
    'avg_transfer': row['AM2trf'],\
    'pub_cost': row['AM2cos']/100}

for row in pm:
    PM[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['PM2dis'], \
    'car_ivt':   row['PM2Tim']/60,\
    'pub_ivt':  row['PM2ivt']/60, \
    'pub_out':  (row['PM2aux']+row['PM2wtt'])/60, \
    'pub_wtt': row['PM2wtt']/60,\
    'pub_walkt': row['PM2aux']/60,\
    'car_cost_erp': row['PM2ERP']/100, \
    'avg_transfer': row['PM2trf'],\
    'pub_cost': row['PM2cos']/100}

for row in op:
    OP[(int(row['origin']),int(row['destin']))]=\
    {'distance': row['OPdis'], \
    'car_ivt':   row['OPTim']/60,\
    'pub_ivt':  row['OPivt']/60, \
    'pub_out':  (row['OPaux']+row['OPwtt'])/60, \
    'pub_wtt': row['OPwtt']/60,\
    'pub_walkt': row['OPaux']/60,\
    'car_cost_erp': row['OPERP']/100, \
    'avg_transfer': row['OPtrf'],\
    'pub_cost': row['OPcos']/100}

for row in zone_employ:
    ZONE[int(row['zone_ID'])]=\
    {'zone_id':int(row['zone_ID']),\
    'zone_code':int(row['zone_code']),\
    'employment':row['employment'],\
    'central_dummy': row['central_dummy'],\
    'parking_rate': row['parking_rate'],\
    'population':row['population'],\
    'area':row['area']}
```

Noted that in the original skims `AMCosts.dat`,`PMCosts.dat`,`OPCosts.dat`, costs are measured in cent and time is measured in minutes. For our variable, we need to first convert them to dollor and hour when generating the dicts (divided by 100 and 60 respectively)

Cost Related Variables

Variables in this section are related to travel cost. Noted that if `origin==destination`, all costs are zero (except parking cost)

```
#public transporation cost for first half tour from origin to destination.
#0 if origin==destination
cost_public_first = AM[(origin,destination)]['pub_cost']
#public transporation cost for second half tour from destination to origin.
#0 if origin==destination
cost_public_second = PM[(destination,origin)]['pub_cost']

cost_bus = cost_public_first + cost_public_second
cost_mrt = cost_public_first + cost_public_second
cost_privatebus = cost_public_first + cost_public_second

#cost of car ERP for first half tour from origin to destination
#0 if origin==destination
```

```
cost_car_ERP_first = AM[(origin,destination)]['car_cost_erp']
#cost of car ERP for second half tour from destination to origin
#0 if origin==destination
cost_car_ERP_second = PM[(destination,origin)]['car_cost_erp']

#cost of car operational cost for first half tour from origin to destination
#0 if origin==destination
cost_car_OP_first = AM[(origin,destination)]['distance']*0.147
#cost of car operational cost for second half tour from destination to origin
#0 if origin==destination
cost_car_OP_second = PM[(destination,origin)]['distance']*0.147

#cost of car parking cost at destination zone (assume an eight hour duration)
cost_car_parking = 8 * ZONE[destination]['parking_rate']

cost_cardriver=cost_car_ERP_first+cost_car_ERP_second+
                cost_car_OP_first+cost_car_OP_second+cost_car_parking
cost_carpassenger=cost_car_ERP_first+cost_car_ERP_second+
                cost_car_OP_first+cost_car_OP_second+cost_car_parking
cost_motor=0.5*(cost_car_ERP_first+cost_car_ERP_second+
            cost_car_OP_first+cost_car_OP_second)+0.65*cost_car_parking

walk_distance1 = AM[(origin,destination)]['distance'] # 0 if origin == destination
walk_distance2 = PM[(destination,origin)]['distance'] # 0 if origin == destination

d1=walk_distance1
d2=walk_distance2

Central_dummy=ZONE[destination]['central_dummy']

cost_taxi_1=3.4+((d1*(d1>10)-10*(d1>10))/0.35+(d1*(d1<=10)+10*(d1>10))/0.4)*0.22+
            cost_car_ERP_first + Central_dummy*3
cost_taxi_2=3.4+((d2*(d2>10)-10*(d2>10))/0.35+(d2*(d2<=10)+10*(d2>10))/0.4)*0.22+
            cost_car_ERP_second + Central_dummy*3

cost_taxi=cost_taxi_1+cost_taxi_2

# the income_mid can be acquired from income class txt file for each income_id
cost_over_income_bus=30*cost_bus/(0.5+Income_mid)
cost_over_income_mrt=30*cost_mrt/(0.5+Income_mid)
cost_over_income_privatebus=30*cost_privatebus/(0.5+Income_mid)
cost_over_income_cardriver=30*cost_cardriver/(0.5+Income_mid)
cost_over_income_carpassenger=30*cost_carpassenger/(0.5+Income_mid)
cost_over_income_motor=30*cost_motor/(0.5+Income_mid)
cost_over_income_taxi=30*cost_taxi/(0.5+Income_mid)
```

Travel Time Related Variables

This section introduces variables related to travel time.

```
# public transportation in vehicle time for first half tour. 0 if `origin==destination`
tt_public_ivt_first = AM[(origin,destination)]['pub_ivt']
# public transportation in vehicle time for second half tour. 0 if `origin==destination`
tt_public_ivt_second = PM[(destination,origin)]['pub_ivt']
# public transportation waiting time for first half tour. 0 if `origin==destination`
tt_public_waiting_first = AM[(origin,destination)]['pub_wtt']
# public transportation waiting time for second half tour. 0 if `origin==destination`
tt_public_waiting_second = PM[(destination,origin)]['pub_wtt']
# public transportation walk time for first half tour. 0 if `origin==destination`
tt_public_walk_first = AM[(origin,destination)]['pub_walkt']
# public transportation walk time for second half tour. 0 if `origin==destination`
tt_public_walk_second = PM[(destination,origin)]['pub_walkt']

# car in vehicle time for first half tour. 0 if `origin==destination`
tt_ivt_car_first = AM[(origin,destination)]['car_ivt']
# car in vehicle time for second falf tour. 0 if `origin==destination`
tt_ivt_car_second = PM[(destination,origin)]['car_ivt']


tt_bus_ivt=tt_public_ivt_first+tt_public_ivt_second
tt_bus_wait=tt_public_waiting_first+tt_public_waiting_second
tt_bus_walk=tt_public_walk_first+tt_public_walk_second
tt_bus_all=tt_bus_ivt+tt_bus_wait+tt_bus_walk

tt_mrt_ivt=tt_public_ivt_first+tt_public_ivt_second
tt_mrt_wait=tt_public_waiting_first+tt_public_waiting_second
tt_mrt_walk=tt_public_walk_first+tt_public_walk_second
tt_mrt_all=tt_mrt_ivt+tt_mrt_wait+tt_mrt_walk

# ! use car ivt !
tt_privatebus_ivt=tt_ivt_car_first+tt_ivt_car_second
tt_privatebus_wait=tt_public_waiting_first+tt_public_waiting_second
tt_privatebus_walk=tt_public_walk_first+tt_public_walk_second
tt_privatebus_all=tt_privatebus_ivt+tt_privatebus_wait+tt_privatebus_walk

tt_cardriver_ivt=tt_ivt_car_first+tt_ivt_car_second
tt_cardriver_out=1.0/6
tt_cardriver_all=tt_cardriver_ivt+tt_cardriver_out

tt_carpassenger_ivt=tt_ivt_car_first+tt_ivt_car_second
tt_carpassenger_out=1.0/6
tt_carpassenger_all=tt_carpassenger_ivt+tt_carpassenger_out
```

```
tt_motor_ivt=tt_ivt_car_first+tt_ivt_car_second
tt_motor_out=1.0/6
tt_motor_all=tt_motor_ivt+tt_motor_out

tt_walk=(d1+d2)/5 #assume the walking speed is 5 km per hour

tt_taxi_ivt=tt_ivt_car_first+tt_ivt_car_second
tt_taxi_out=1.0/6
tt_taxi_all=tt_cardriver_ivt+tt_cardriver_out

average_transfer_number = (AM[(origin,destination)]['avg_transfer'] +
PM[(destination,origin)]['avg_transfer'])/2
```

Other Variables

```
# this data is in 2008 planning data by LTA 1092 MTZ Zones.xls

resident_student = ZONE[origin]['resident students']
education_op = ZONE[destination]['total enrolment']

origin_area= ZONE[origin]['area'] # in square km
destination_are = ZONE[destination]['area'] # in square km

residential_size=resident_student/origin_area/10000.0
school_attraction=education_op/destination_area/10000.0


# The following dummy variables are generated based on
# household car, motorcycle ownership.

zero_car = 1*(car_own_normal==0)
one_plus_car = 1*(car_own_normal>=1)
two_plus_car = 1*(car_own_normal>=2)
three_plus_car = 1*(car_own_normal>=3)

zero_motor = 1*(motor_own==0)
one_plus_motor = 1*(motor_own>=1)
two_plus_motor = 1*(motor_own>=2)
three_plus_motor = 1*(motor_own>=3)

age_over_15 = 1 * (age_id >= 3)
university_student = 1 * (student_type_id == 6)
```

Availability of Alternatives

public bus, private bus and MRT/LRT are only available `if AM[(origin,destination)]['pub_ivt']>0 and PM[(destination,origin)]['pub_ivt']>0:`

shared2, shared3+, taxi and motorcycle are available to all.

Walk is only avaiable `if (AM[(origin,destination)]['distance']<=5 and PM[(destination,origin)]['distance']<=5)`

drive alone is available when for the agent, `has_driving_license * one_plus_car == True`