A short guide for using the MATLAB code for estimating the recursive logit model

Université de Montréal

September 13, 2015

This document provides a brief introduction to the use of the MATLAB code for estimating the recursive logit (RL) model (Fosgerau et al., 2013).

1 Data structure

In order to estimate the RL model we need to provide the code with the network structure, the link attributes (attributes of a link given another links) and trip observations. The network structure (including links and connections between links) and link attributes are stored by incident matrices.

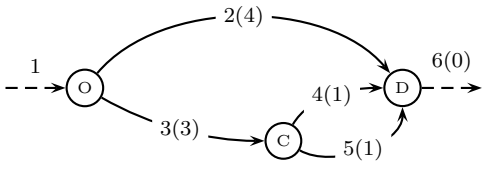

Figure 1: Classic three paths example network

For example, we consider the three paths network shown in Figure 1. There are 6 links which are numbered as 1,2,3,4,5 and 6. The values in parentheses refer to the link lengths. Note that link 6 is the dummy link which has no successor added to the destination D (see for instance Fosgerau

et al., 2013), the corresponding incidence matrix is

$$\text{Incidence} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Also, the link lengths are also represented as a matrix of the same size

$$\text{link lengths} = \begin{pmatrix} 0 & 4 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Indeed, these matrices are very sparse, so it is convenient to store them in sparse structure. The following commands are often used to convert between a full matrix and a sparse matrix: *sparse()* and *spconvert()*.

Path observations are requires for estimating the model. They also have to be stored in a matrix where each row corresponds to a path. On each row, the first element is the respective destination followed by a sequence of links from the origin to destination. For example, row (6,1,3,4,6,0..) represents path (1,3,4,6).

2 Using the code

The main file is "*RLoptimizer.m*". First, we need to provide the code with the data files, i.e., the files contain link incidence, link attributes and path observations.

```
file_linkIncidence = './Input/linkIncidence.txt';
file_AttEstimatedtime = './Input/ATTRIBUTEestimatedtime.txt';
file_turnAngles = './Input/ATTRIBUTEturnangles.txt';
file_observations = './Input/SyntheticObservations.txt';
```

Note that these files must contain matrices in sparse structure.

Second, we need to load the data by using file "*loadData.m*". In this file, variable *isLinkSizeInclusive* is assigned the *true* value if we want to include the Link Size attribute (For instance Fosgerau et al., 2013) to the RL model. The command *Atts = getAtt();* is used to construct the attribute vectors. The number of attributes we use is also the number of objects in *Atts*. These objects are defined in file *getAtt.m*.

Third, the number of parameters to be estimated is set in file *initialize-optimization-structure.m* (variable *Op.n*). In this file we also can choose an initial parameters for the optimization algorithm.

Fourth, the optimization algorithm can be specified in the main file as

```
Op.Optim_Method = OptimizeConstant.TRUST_REGION_METHOD; %
    Optimization algorithm
Op.Hessian_approx = OptimizeConstant.BFGS; % Hessian approximation
```

Typically, the trust region method with BFGS is efficient to use. Now the model is ready to be estimated.

References

M. Fosgerau, E. Frejinger, and A. Karlström. A link based network route choice model with unrestricted choice set. *Transportation Research Part B*, 56:70–80, 2013.