Comparison of path- and link-based choice models for dynamic traffic assignment*

Wilco Burghout[†] and Emma Frejinger[‡]

March 27, 2013

Extended abstract for hEART symposium 2013

*Research funded by TRENoP strategic research area.
†Centre for Traffic Research, KTH Royal Institute of Technology
‡DIRO, Université de Montréal, Canada

Considering the increasing traffic problems due to congestion, dynamic traffic assignment (DTA) models are important in the analysis, planning and management of transport networks. Predicting route choices is central in DTA models (e.g. DYNAMEQ, Florian et al., 2001, DYNASMART, Jayakrishnan et al., 1994), DYNAMIT, Ben-Akiva et al., 2002 and Mezzo, Burghout et al., 2006, Burghout et al., 2010) and for this purpose path based logit models are generally used. Such models require sampling of choice sets of paths which is usually done by iteratively computing shortest paths using different generalized costs. Frejinger et al. (2009) show that for consistent estimation of model parameters the path utilities should be corrected for the sampling protocol. It is however still unclear how to correct the utilities when the models are used for prediction, which is the case in DTA models. Recently Fosgerau et al. (2012) proposed the recursive logit (RL) model which is a link-based choice model that requires no restriction on the choice set and no path sampling. Moreover, it can be both consistently estimated and efficiently used for prediction.

The objective of this paper is to compare the RL and path-based logit models for DTA with regard to computational complexity, memory requirements and prediction accuracy. We provide a theoretical discussion as well as numerical results using the DTA model Mezzo (Burghout et al., 2006) for an illustrative example and for a real network.

Recursive logit model

In this section we briefly describe the RL model which corresponds to a logit model over an unrestricted choice sets of paths. The choice of path is decomposed into a sequence of link-choices and is formulated as a dynamic discrete choice model Rust (1987). The choice of next link a (action) is made a the sink node of the current link k (state). An instantaneous utility is associated with each link $u(a|k) = v(a|k) + \mu(\varepsilon(a) - \gamma)$ where $v(a|k)$ is the deterministic utility that is negative by definition for all links except the dummy link without successors representing the destination. Similar to a classic logit model $\varepsilon(a)$ are i.i.d. extreme value. A traveller chooses the link for which the sum of the instantaneous utility and the expected utility until the destination $V(a)$ is the maximum. In the context the expected maximum utility (also referred to as the value function) is the log sum and is defined by the recursive Bellman equation $V(k) = E\left[\max_a(v(a|k) + V(a) + \mu\varepsilon(a))\right]$.

Let $\mathbf{M}$ denote the incidence matrix defining instantaneous utilities with elements $M_{ak} = \delta(a|k)e^{\frac{1}{\mu}v(a|k)}$ if a is incident to k and zero otherwise. We denote $\mathbf{z}$ a vector containing the value functions for each link $z_k = e^{\frac{1}{\mu}v(a|k)}$ and $\mathbf{b}$ a vector defining the destination (elements for all links zero except the one for the destination which equals one). The Bellman equations can then be written as a system of linear equations

$$\mathbf{z} = \mathbf{M}\mathbf{z} + \mathbf{b} \tag{1}$$

which can be efficiently solved. Note that, this formulation is valid also for time dependent (deterministic) link travel times if a state is defined as a link time pair instead of a link. We organize the next link probabilities into a matrix $\mathbf{P}$ where a row k is defined as $\mathbf{P}_k = \frac{\mathbf{M}_k \circ \mathbf{z}^T}{\mathbf{M}_k \mathbf{z}}$ and where $\circ$ is the element-by-element product.

Theoretical comparison

In this section we compare the link- and path-based approaches by discussing memory requirements and algorithm complexity (without a formal complexity analysis). The objective of this section is to highlight the differences between the two approaches rather than drawing general conclusions. We start by discussing the differences in choice set assumptions.

Any path, including those with loops if the network contains cycles, is included in the choice set of the RL model. The corresponding choice set of paths is hence infinite. This is fundamentally different from existing DTA path choice models where the choice sets in general include only a small number of paths. The path sampling approach relies on the same assumption as the RL model but it is not known how to correct path utilities and the sampling approach is therefore not applicable in a DTA model. Even though paths with loops should have a choice probability close to zero for a well specified model, it needs to be numerically investigated whether an unrestricted choice set cause issues in a DTA model. It is also clear that restricting the choice set to too few paths may also be undesirable.

We now turn our attention to the comparison of memory requirements. We assume that the path-based approach uses static choice sets but with time-dependent path utilities. The link-based approach has however time-dependent alternatives. In spite of this difference which favours the path-based approach we make an approximative comparison. Note that, there are many different network related attributes that need to be stored that are common to the two approaches, below we focus on the differences.

The memory the path-based approach requires depends on the size of the choice sets and the number of links in a path. On average, considering static choice sets $|S||C|M_p(\bar{I})$ memory is needed where $|S|$ is the number of OD pairs, $|C|$ is the average choice set size ($|C| = \frac{1}{|S|}\sum_{s\in S}|C_s|$) and $M_p(\bar{I})$ the memory required for storing a path with average number of links $\bar{I}$. This is a crude approximation and should mostly serve as an indication of what factors influence the memory requirement. We note that for a given OD pair, paths often share links but it is difficult to exploit this redundancy to reduce the memory requirement since overlapping sequences may occur at different locations in the path.

In order to compute the next-link probabilities using the RL model, the destination specific value functions are stored. The memory required is then approximately $|A||T||D|M_f$, where $|A|$ is the number of links, $|T|$ the number of time periods, $|D|$ the number of destinations and M_f the memory required for a float. If the OD matrix is square then $|D| = \sqrt{|S|}$. On the one hand, even for paths with few links $M_p(\bar{I}) >> M_f$, but $|A||T| >> |C|$. On the other hand, the link based approach is destination specific whereas the choice sets are OD specific.

We now turn our attention to a brief informal discussion on algorithmic complexity. The computational complexity of the path based approach mainly stems from the repeated path generation. Indeed, shortest path trees are computed repeatedly and the path structure is traversed a number of times for checking uniqueness, calculating costs and pruning infeasible or expensive paths. The logit choice probabilities are on the other hand straightforward to compute.

The RL model requires the solution of a system of linear equations for each destination. The time complexity of solving such a system (if it is small enough to be solved with a direct method) is comparable to the the complexity of solving one shortest path tree. The solution to the system depends on the link travel times and hence needs to be solved for each assignment, but it is done once for each destination. The destination specific next link probabilities require $|A||T|$ element-by-element vector multiplications and one matrix multiplication.

Final remarks

This is ongoing work and we are currently finalizing the Mezzo implementation of the RL model. In order to compute (1) efficiently we use a MATLAB bridge to Mezzo. In the paper, we will present numerical results on an illustrative example in order to highlight differences of the model. This small network is composed of 12 links and one OD pair. We choose link lengths and capacities so that there are four comparable path alternatives and we analyze the results for different demands. The real data is from the Södermalm network which is a part of Stockholm, it is composed of 1101 links and 577 nodes and the path version of Mezzo has already been calibrated on this network. We compare the assignment results for the two different models and for different settings of Mezzo. For validation purposes there are two different sources of data: link counts and observed path choices from taxi GPS data.

The main contribution of this paper is a discussion on these two fundamentally different route choice models from theoretical and numerical points of views. We analyze the paths choices generated by the models as well as the requirements concerning memory and computational resources. The RL model has clear theoretical advantages (can be consistently estimated and used for prediction without utility correction) and we analyze here potential practical issues such as the presence of loops.

References

Aguirregabiria, V. and Mira, P. (2010). Dynamic discrete choice structural models: A survey, *Journal of Econometrics* **156**(1): 38–67.

Akamatsu, T. (1996). Cyclic flows, markov process and stochastic traffic assignment, *Transportation Research Part B: Methodological* **30**(5): 369–386.

Baillon, J.-B. and Cominetti, R. (2008). Markovian traffic equilibrium, *Mathematical Programming* **111**(1-2): 33–56.

Bell, M. (1995). Alternatives to dial's logit assignment algorithm, *Transportation Research Part B: Methodological* **29**(4): 287–295.

Bellman, R. (1957). *Dynamic Programming*, Princeton University Press, Princeton.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. and Mishalani, R. (2002). Real time simulation of traffic demand-supply interactions within DynaMIT, *Transportation and network analysis: current trends. Miscellenea in honor of Michael Florian*, Kluwer Academic Publishers, pp. 19–36.

Burghout, W., Koutsopoulos, H. and Andreasson, I. (2006). A discrete-event mesoscopic traffic simulation model for hybrid traffic simulation, *Proceedings of the Intelligent Transportation Systems Conference*, Toronto, Canada.

Burghout, W., Koutsopoulos, H. and Andreasson, I. (2010). Incident management and traffic information: Tools and methods for simulation-based traffic prediction, *Transportation Research Record* **2161**: 20–28.

Dial, R. (1971). A probabilistic multipath traffic assignment algorithm which obviates path enumeration, *Transportation Research* **5**(2): 83–111.

Florian, M., Mahut, M. and Tremblay, N. (2001). A hybrid optimization-mesoscopic simulation dynamic traffic assignment, *IEEE Intelligent Transportation Systems Conference Proceedings*, Oakland (CA), USA.

Fosgerau, M., Frejinger, E. and Karlström, A. (2011). A logit model for the choice among infinitely many routes in a network, *Technical report*, Center for Transport Studies, KTH.

Frejinger, E., Bierlaire, M. and Ben-Akiva, M. (2009). Sampling of alternatives for route choice modeling, *Transportation Research Part B* **43**(10): 984–994.

Gao, S., Frejinger, E. and Ben-Akiva, M. (2008). Adaptive route choice models in stochastic time dependent networks, *Transportation Research Record* **2085**: 136–143.

Jayakrishnan, R., Mahmassani, H. and Hu, T. (1994). An evaluation tool for advanced traffic information and management systems in urban networks, *Transportation Research Part C* **2**(3): 129–147.

Rust, J. (1987). Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher, *Econometrica* **55**(5): 999–1033.