Logit route choice and stochastic user equilibrium

CE 392C

This is the third variation on the basic equilibrium problem we'll see.

In practice, travelers do not always pick the least-travel time path between their origin and destination. Why?

Two possible reasons are

1. People care about factors other than travel time;
2. People don't know the travel times on all routes accurately.

Interestingly, both interpretations lead us to the same place mathematically (under some assumptions).

We can express these ideas using the concepts of “observed” and “unobserved” utility:

$$U_i = V_i + \epsilon$$

where ϵ is a random variable. The unobserved utility U_i is the actual (private) level of satisfaction a traveler would get by making choice i; V_i is the component of this utility that us (the planner) has access to.

Under the “charitable” interpretation, ϵ reflects other factors influencing route choice that we are not directly modeling. Under the “uncharitable” interpretation, ϵ reflects the perception error in travel times.

Specifically, for a path π, its utility can be expressed

$$U_\pi = -c_\pi + \epsilon_\pi$$

and each traveler will choose the path with the highest value of U_π for him or her.

Depending on the assumptions made on the distribution of ϵ_π, we obtain different route choice models. All of these are called *stochastic network loading* models, because of the random terms in the utility function.

If we assume that the ϵ are independent, identically distributed random variables with zero mean, and that each ϵ_π has a Gumbel distribtion, then we obtain the **logit** route choice model.

If we assume that the ϵ are (possibly correlated) normal random variables, we obtain the **probit** model.

The logit model is much simpler and easier to work with, but has a few shortcomings (which will be explored on the homework). The probit model is more realistic, but becomes very complicated once there are more than a handful of routes to choose from.

Then, according to the logit formula, the proportion of drivers traveling between origin r and destination s who choose path $\pi \in \Pi^{rs}$ is

$$p_{rs}^{\pi} = \frac{\exp(-\theta c^{\pi})}{\sum_{\pi' \in \Pi^{rs}} \exp(-\theta c^{\pi'})}$$

where θ reflects the importance of the random term relative to the travel time (it is related to the variance of ϵ).

Example

Three routes connect an OD pair; their travel times are 10, 15, and 20 minutes and do not depend on traffic flow. If the demand for this OD pair is 1000 vehicles and $\gamma = 0.1$, how many vehicles choose each route?

Two challenges with this approach:

- Do travelers really consider *all* paths between an origin and destination?
- Is there a way to avoid enumerating paths?

Enter Dial's method.

DIAL’S METHOD

Dial's method involves a definition of *reasonable paths* (a subset of all paths in the network), and then shows us how to identify the link flows *without* having to list off any individual paths.

Reasonable paths

Consider all of the travelers departing origin r, and let L_i be the travel time on the shortest path from r to every other node i.

A link (i, j) is a *reasonable link* if $L_i < L_j$. (That is, it takes you "away" from the origin.)

A path is a *reasonable path* if it is comprised only of reasonable links. (It never doubles back toward the origin.)

There are other definitions of reasonable paths as well. Can you think of some?

For a given origin, the set of reasonable links implies a *bush*.

For a network $G = (N, A)$, a *bush* $\mathcal{B}$ is a set of links satisfying these conditions:

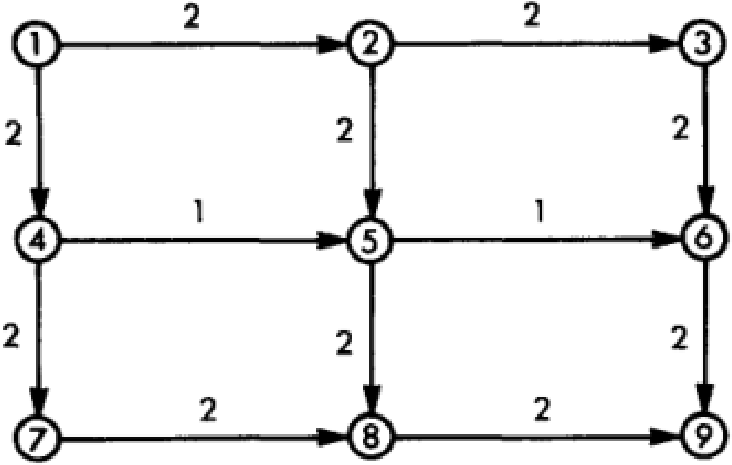
- $\mathcal{B} \subseteq A$ (The bush links all are part of the original network).
- There is a "root" node r, such that every node reachable from r in the original network is also reachable from r only using bush links. (The bush is connected to the root.)
- The collection of links in B has no cycles. (This is the key property. Acyclic networks make our lives easy.)

In short, a bush is a acyclic subnetwork which is "connected" in the sense that you can reach any other node from the root.

In particular, for an origin r, the set of all reasonable links implies a bush rooted at r.

We assume that these are the only links that travelers leaving node r will use.

Example



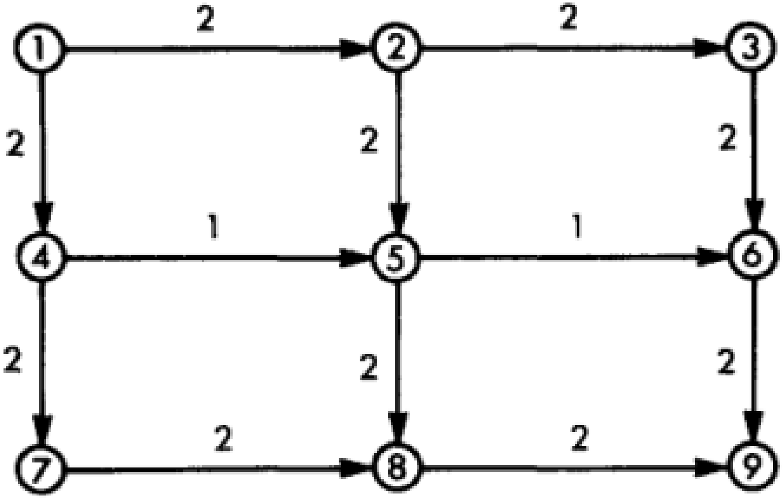
Enumerating reasonable paths, what should the link flows be?

The second part of Dial's method is a way to determine the link flows *without* having to list off every path ahead of time.

Perform the following steps (only considering the reasonable bush).

1. Calculate the *link likelihood* $L_{ij} = \exp(\theta(L_j - L_i - t_{ij}))$
2. Going in forward topological order, calculate the weights W_{ij} for links leaving node i:
   1. For each node i, the "total incoming weight" is $W_i = \sum_{(h,i)\in A^B} W_{hi}$ (or 1 if i is the origin).
   2. Distribute this weight to the outgoing links: $W_{ij} = W_i L_{ij}$
3. Going in reverse topological order, calculate the flows x_{ij} for links entering node j:
   1. For each node j, the total flow passing through node j is $X_j = q_{rj} + \sum_{(j,k)\in A} x_{jk}$.
   2. Distribute this flow on the incoming links proportional to the weights: $x_{ij} = X_j W_{ij}/W_j$.

Example



An intuitive understanding of these concepts:

The *link likelihood* lets us deal with choices at the link level, rather than the path level:

$$L_j - L_i - t_{ij}$$

represents "how close" (i, j) is to being on the shortest path to j; this quantity is zero if (i, j) on a shortest path to j, and strictly negative otherwise.

Furthermore, by Bellman's principle, this is also a measure of "how close" (i, j) is to being on shortest paths *to any node topologically following* j.

So, $\exp(\theta(L_j - L_i - t_{ij}))$ expresses the relative likelihood of using link (i, j).

The *link weight* is related to the link likelihood, but also takes into account *the number of reasonable paths that use this link*.

(The logit model assumes each path is independent of every other; so the more paths that use a link, the more flow it should have.)

To get the link weight, we multiplied the link likelihood by the node weight (the total weight on incoming links).

Because the reasonable links form a bush, we can start at the root (origin) and work forwards in topological order.

The *link flows* are calculated by assigning flows proportional to the link weights.

More specifically, consider a single node, and assume that we know how many vehicles pass through this node (either because it is their destination, or because they are traveling on one of the links leaving this node).

These vehicles are then distributed on the upstream links according to the link weights.

Because the reasonable links form a bush, we can start at the "end" of the network and work backwards in topological order.

What if there is more than one origin?

Simply apply Dial's method to each origin in turn. Each origin has its own bush, corresponding to its own shortest path labels and reasonable links.

Add up the link flows from each origin to get the total link flows.

Since travel times are constant, it doesn't matter what order we load origins in.

STOCHASTIC LOADING TO STOCHASTIC EQUILIBRIUM

Dial's method assumes that the link travel times are fixed. (This is the *stochastic network loading* problem.)

We want to handle cases where link travel times depend on flows, using link performance functions. (This is the *stochastic user equilibrium* problem.)

Remember, the principle of stochastic user equilibrium is

> Every traveler chooses the path which minimizes his or her *perceived* travel time.

It turns out that the stochastic user equilibrium can be seen as the solution to a convex optimization problem, to a variational inequality, or to a fixed-point problem.

The convex objective function is:

$$\min_{\mathbf{x},\mathbf{h}} \sum_{(i,j)\in A} t_{ij}(x)\ dx + \frac{1}{\theta} \sum_{\pi\in\Pi} h^\pi \log h^\pi$$

subject to the usual constraints.

It is also possible to express this function in terms of origin-disaggregated link flows (but *not* total link flows).

To see why, first note that the non-negativity condition $h \geq 0$ can be disregarded (why?) and write the optimality conditions for this problem:

$$\frac{\partial \mathcal{L}}{\partial h^\pi} = c^\pi + \frac{1}{\theta}(\log h^\pi + 1) + \kappa^{rs} = 0$$

Solving for h^π and using the demand constraint gives:

$$h^\pi = d^{rs} \exp(-\theta c^\pi) / \sum_{\pi' \in \hat{\Pi}^{rs}} \exp(-\theta c^{\pi'})$$

which is exactly the logit equation.

The VI can be formulated directly from the definition of a minimum to the convex program.

For the fixed-point formulation, let $\mathbf{c}(\mathbf{h})$ be the vector of path travel times as a function of the vector of path flows.

Now, given the travel time on each path, the flow on each path is given directly by the logit equation:

$$h^\pi = d_{rs} p^\pi_{rs} = d_{rs} \frac{\exp(-\theta c^\pi)}{\sum_{\pi' \in \hat{\Pi}^{rs}} \exp(-\theta c^{\pi'})}$$

Let $\mathbf{H}(\mathbf{c})$ be the vector version of this formula.

Then the SUE problem can be expressed as follows: find $\mathbf{h} \in H$ such that $\mathbf{h} = \mathbf{H}(\mathbf{c}(\mathbf{h}))$.

Since both $\mathbf{H}$ and $\mathbf{c}$ are continuous functions, a stochastic user equilibrium always exists.

This is much easier than before, since the path flows are *uniquely determined* from the path travel times.

One downside of fixed-point problems is that there is no clear solution algorithm available.

However, since a convex objective function does exist, MSA and Frank-Wolfe are guaranteed to work.

MSA can be used easily; FW is harder since calculating the derivatives of the objective function requires path enumeration.

To do this, we use Dial's method to calculate our $\mathbf{x}^*$ in the usual iterative process:

1. Choose an initial feasible assignment $\mathbf{x}$.
2. Update travel times.
3. Calculate target flows $\mathbf{x}^*$.
4. Adjust flows from $\mathbf{x}$ in the direction of $\mathbf{x}^*$.

In other words, rather than only putting travelers on a single path in $\mathbf{x}^*$, in SUE vehicles are loaded on multiple paths, using Dial's method with the current link travel times.

MORE GENERAL SUE MODELS

Rather than using Dial's definition of reasonable paths, we can use other definitions of which paths are allowed.

Logit assignment can also be done conveniently if literally *all* paths are allowed (even those with cycles).

Logit assignment can be used with any definition of allowable paths, but there may not be an efficient way to calculate the resulting link flows. (The logit formula can always be directly applied.)

Probit models can also be used, but it is harder both to (1) do stochastic network loading to get link flows; and to (2) formulate an objective function for minimization.

In practice, probit models are solved using Monte Carlo sampling.