The 21st Summer Course

"Behavior Modeling in Transportation Networks"

Lecture Series 1 (16:00-16:30 Sep. 23, 2022)

Recursive structure of decision making in networks: Modeling and estimation

Yuki Oyama

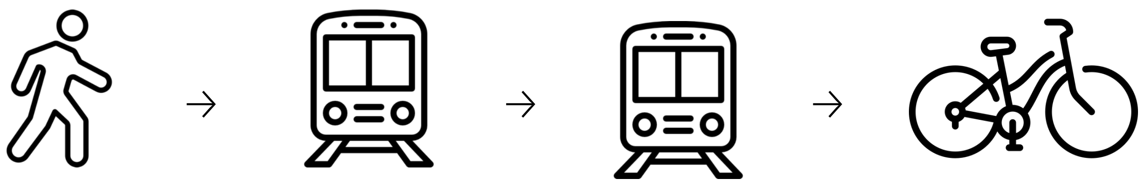
Shibaura Institute of Technology

Activity Landscape Design Lab.

oyama@shibaura-it.ac.jp

Sequential choice problem

Ex.) A trip is completed by using a **sequence of multiple modes**:



Travel behavior decisions often involve sequential choices

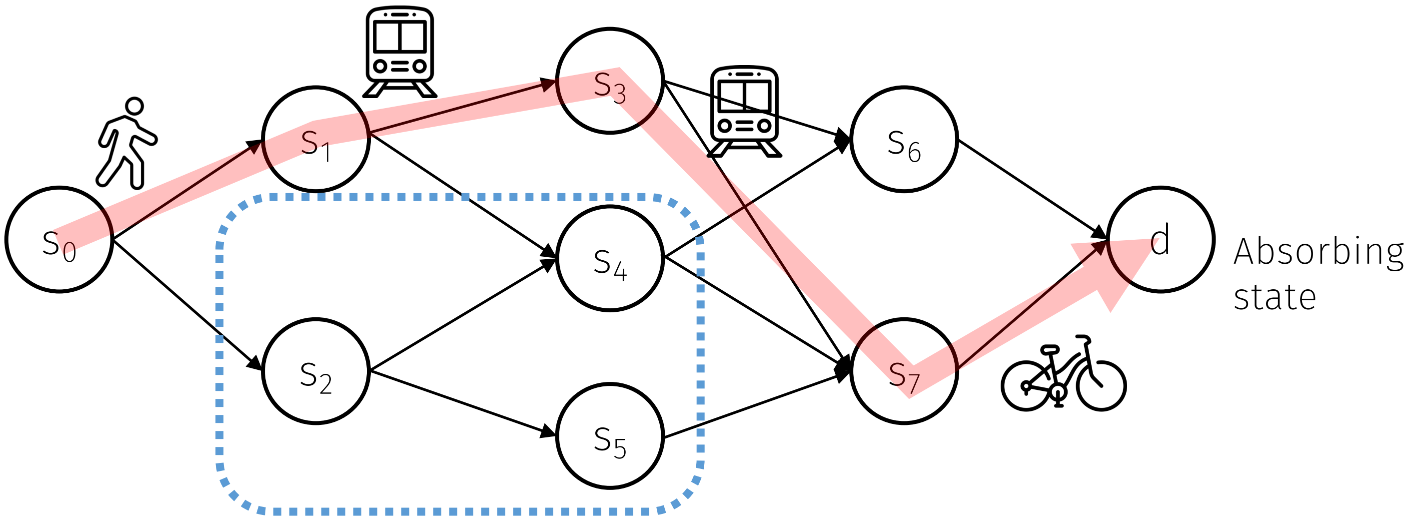
- **Mode choice** (Walk – Train 1 – Train 2 - Bike)
- **Activity pattern** (Home – Work – Other – Other - Home)
- **Spatial movement** (Place 1 – Place 3 – Place 4 – Place 9)
- **Timing decision** (Stay – Stay – Stay – Action)

Challenges:

1. Hard/impossible to enumerate available sequences (**Computation**)
2. Travelers may not perceive all elements in a sequence at once (**Decision-making structure**)

Sequential path choice in a state network

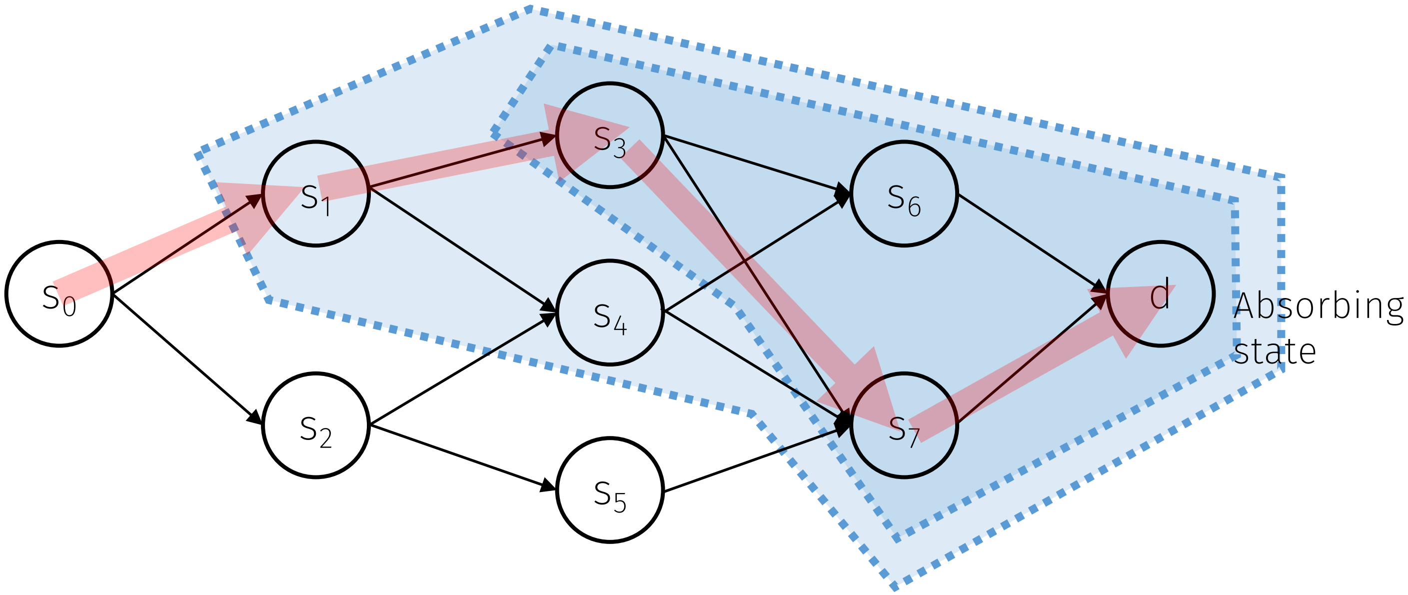
Define choice elements as ***state***s and their **network**



- A sequential choice is described as a path in the network
- An elemental choice corresponds to a substructure of network whose choice set is tractable (easy to count)
    - E.g., the choice set available to S_2 is $\{S_4, S_5\}$

Sequential path choice in a state network

Define choice elements as ***state***s and their **network**

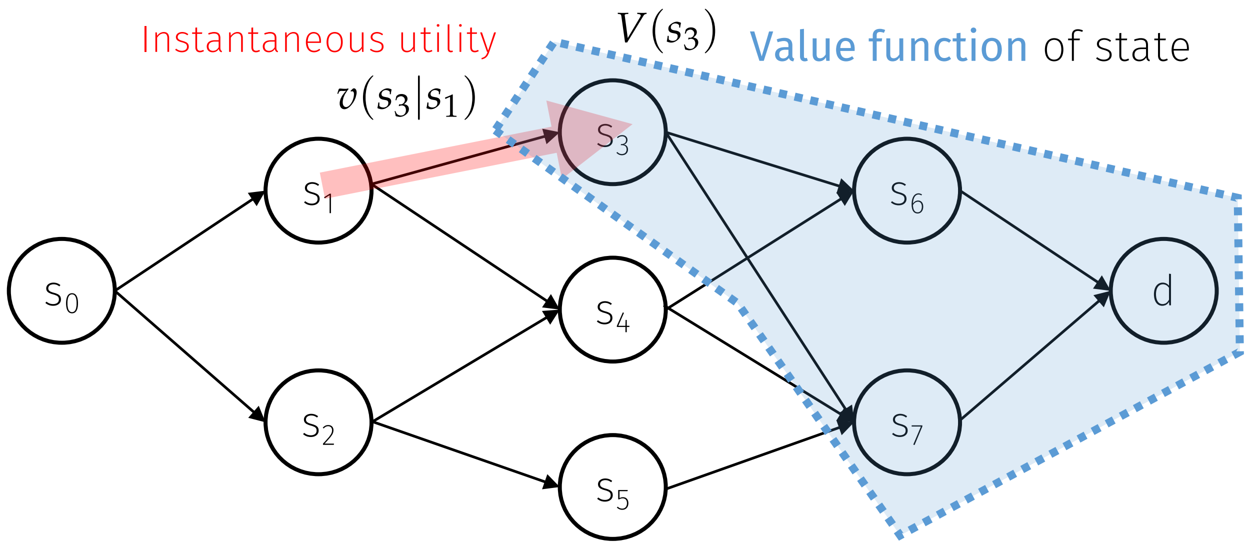


Recursive structure of decision-making:

- Not direct choice of a sequence, but sequential choices of states
  - Elemental choice is the choice of next state given the current state
- When choosing a state, travelers consider their possible future choices
  - Trade-off between the current stage utility and the future expected utility

Sequential path choice in a state network

Define choice elements as ***state***s and their **network**



Recursive logit (RL) model (i.e., deterministic MDP with i.i.d. Gumbel utilities)

$$V^d(k) \equiv \mathbb{E}\left[\max_{a \in A(k)} \{v(a|k) + V^d(a) + \mu\epsilon(a|k)\}\right] \quad \Rightarrow \quad \underbrace{e^{\frac{1}{\mu}V^d(k)}}_{z_k} = \sum_{a \in A(k)} \underbrace{e^{\frac{1}{\mu}\{v(a|k)+V^d(a)\}}}_{M_{ka} z_a}$$

$$\boldsymbol{z} = \mathbf{M}\boldsymbol{z} + \boldsymbol{b} \Leftrightarrow \boldsymbol{z} = (\mathbf{I} - \mathbf{M})^{-1}\boldsymbol{b}$$ System of linear equations

Dynamic decision-making description & Efficient computation

Recursive model's feasibility

The value functions are the solution of a **fixed point problem**:

$$z = \mathcal{T}_\beta(z) \quad \text{where} \quad \mathcal{T}_\beta(z) \equiv \mathbf{M}z + b \quad \text{is the Bellman operator}$$

1. Hawkins-Simon / spectral radius condition

$(\mathbf{I} - \mathbf{M})$ is **invertible** when the maximum absolute of eigenvalues of $\mathbf{M}$ is strictly less than one $\rho(\mathbf{M}) < 1$

2. Utility condition (Mai and Frejinger, 2022)

$\mathcal{T}_\beta(z)$ is a **contraction mapping** if:

$$\sum_{a \in A(k)} M_{ka} < 1, \forall k \in A \qquad \text{where} \quad M_{ka} \equiv e^{\frac{1}{\mu} v(a|k)}$$

(The nested RL (NRL) case)

$$\sum_{a \in A(k)} M_{ka}\phi_{ka} < 1, \forall k \in A \qquad \text{where} \quad \phi_{ka} \equiv \mu_a / \mu_k$$

Model estimation with recursive structure

Nested Fixed Point (NFXP) algorithm (Rust, 1987)

Outer algorithm

maximizes the likelihood function over the parameter space

$$\max_{\beta} LL(\beta, \boldsymbol{z}(\beta)) \qquad \text{e.g.} \quad \beta^{(m+1)} \leftarrow \beta^{(m)} - \mathbf{H}^{(m)} \frac{\partial LL(\beta^{(m)})}{\partial \beta}$$

Inner algorithm

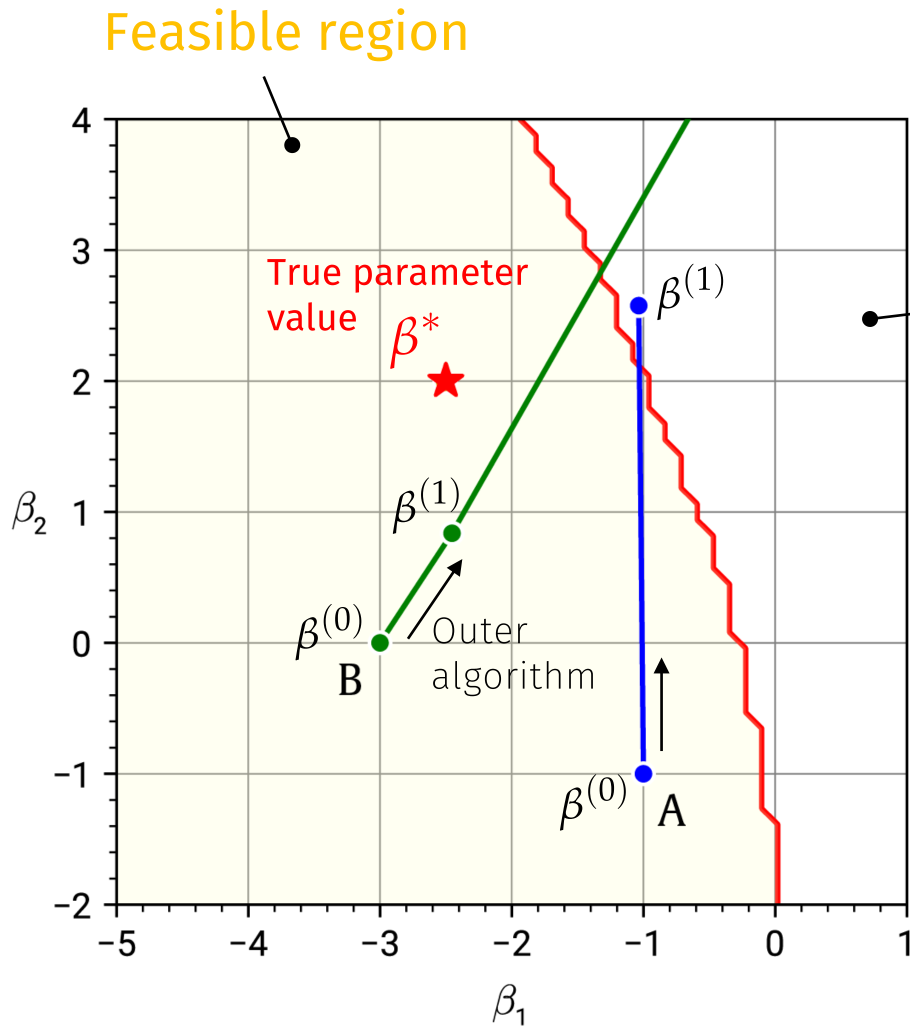
solves the value functions for each parameter value

$$\boldsymbol{z} = \mathcal{T}_{\beta^{(m)}}(\boldsymbol{z}) \qquad \boldsymbol{z} = (\mathbf{I} - \mathbf{M})^{-1}\boldsymbol{b} \quad \text{or} \quad \boldsymbol{z}^{(l+1)} \leftarrow \mathbf{M}\boldsymbol{z}^{(l)} + \boldsymbol{b}$$

(until convergence)

Inner algorithm **must be feasible for ALL parameter values** $(\beta^{(0)}, \beta^{(1)}, \dots)$ searched during the estimation

Why is the issue more critical during estimation?



Infeasible region

$$z = \mathcal{T}_{\beta^{(m)}}(z)$$

does not have a fixed point solution

Estimation problem of recursive models

1. Even if the true and initial parameter values satisfy the feasibility condition, **parameter values during the estimation process can often violate the condition** (and the estimation fails).

2. Moreover, unknown **true parameter does not necessarily satisfy the condition**.

E.g., if there exist state pairs with **positive utilities**

$$M_{ka} = e^{v(a|k)} > 1, \text{ if } v(a|k) > 0$$

Then the following condition clearly does **NOT** hold:

$$\sum_{a \in A} M_{ka} < 1, \ \ \forall k \in A$$

What have been done

Ad hoc manipulation of utility function

To somehow satisfy the utility conditions, previous studies

- Include **ONLY negative attributes** (e.g., travel time, turn penalties)
- Add a **fixed large penalty** term
- Start estimation with a parameter value with large negative magnitude

Two problems:

1. **Limit the practical applicability of RL models** (cannot capture positive network attributes)
2. This manipulation **does not ensure a solution or stable estimation**.

This study proposes the use of

Prism-based approach (Oyama and Hato, 2019)

The idea of prism-based approach

To interpret, the value function can be viewed as:

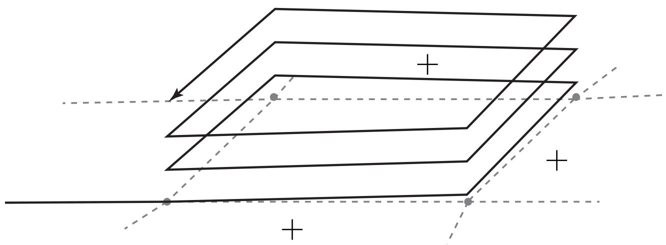
$$V^d(k) = \mathbb{E}\left[\max_{r \in \mathcal{R}_{kd}} \{v(r) + \mu\epsilon(r)\}\right] = \ln \sum_{r \in \mathcal{R}_{kd}}^{\infty} e^{\frac{1}{\mu} v(r)}$$

Expectation of **ALL path utilities** connecting state k to destination d

$$\boldsymbol{z} = (\mathbf{I} - \mathbf{M})^{-1}\boldsymbol{b} = (\mathbf{I} + \mathbf{M} + \mathbf{M}^2 + \cdots)\boldsymbol{b}$$

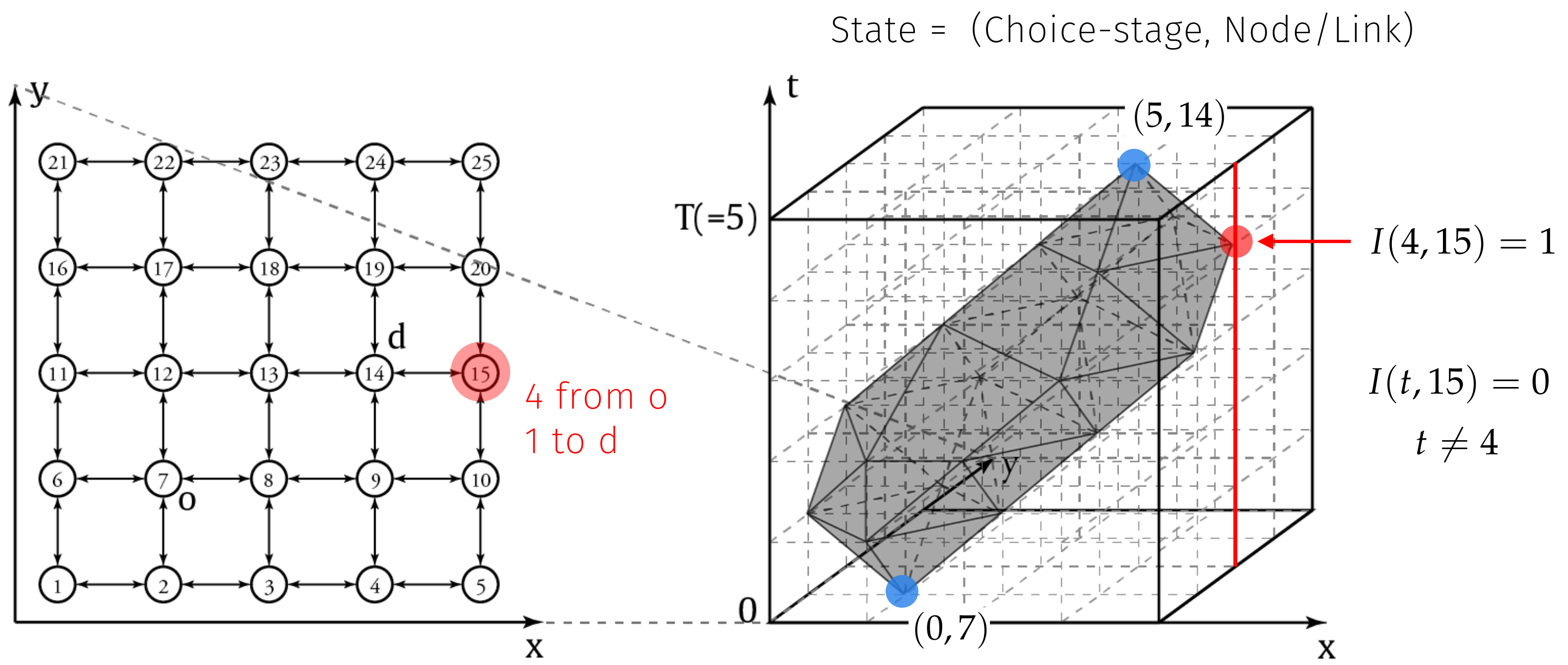
Power series as **implicit calculation process of the path utilities**

ex.) Unrealistic cycles that explosively gain utilities



Prism-based approach **efficiently and behaviorally restricts unrealistic paths** that cause the numerical issue

Prism-based path set restriction



1. Define a state-extended network based on **choice-stage**
2. Define the **choice-stage constraint** *T*, and evaluate **state existence conditions** $I(t, k)$ based on the minimum number of steps from *o* and to *d*
3. **States connection condition**: $\Delta_t(a|k) = I(t, k)\delta(a|k)I(t + 1, a)$
4. The reduced set of states **forms into a prism**.

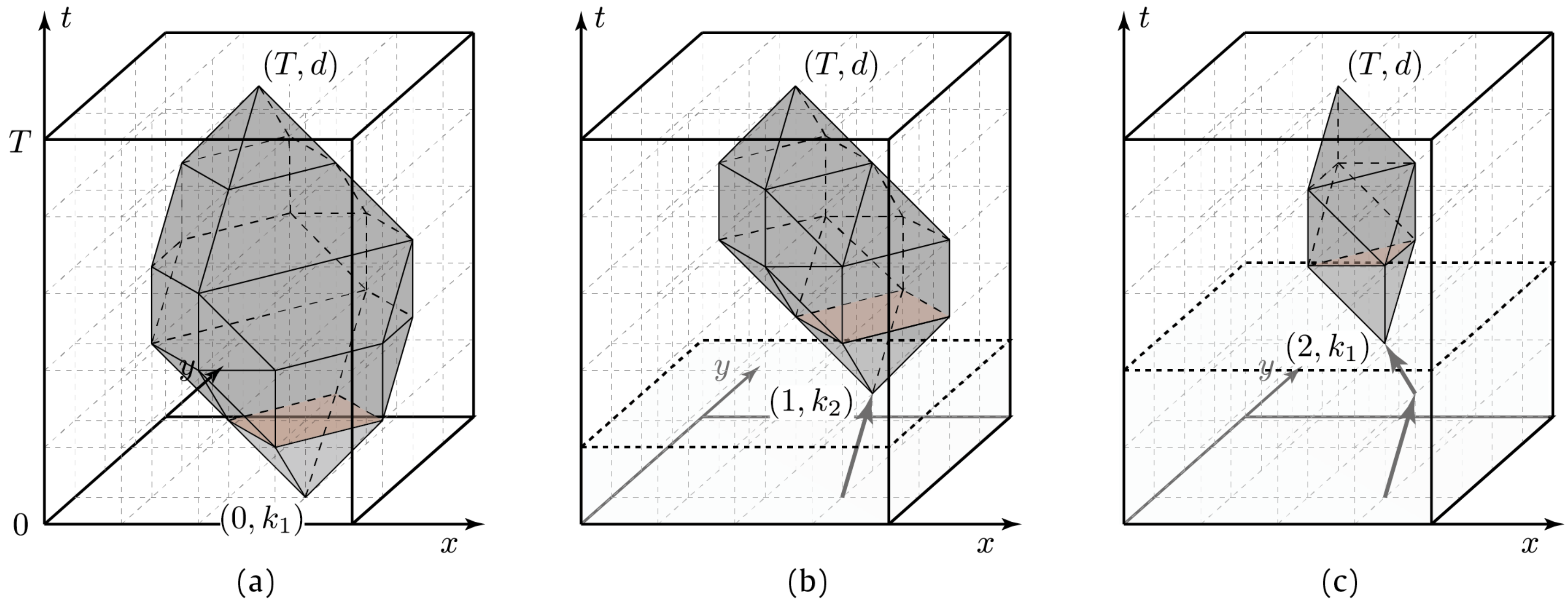
Prism-constrained RL model

$z_{t,k} \equiv e^{\frac{1}{\mu}V(t,k)}$

$M'_{t,ka} \equiv \Delta_t(a|k)e^{\frac{1}{\mu}v(a|k)}$

Redefine the value function $V(t,k)$ for each state:

$$e^{\frac{1}{\mu}V^d(t,k)} = \sum_{a\in A(k)} \Delta_t^d(a|k) e^{\frac{1}{\mu}\{v(a|k)+V^d(t+1,a)\}} \quad \Leftrightarrow \quad \boldsymbol{z}_t^d = \mathbf{M}_t'^d \boldsymbol{z}_{t+1}^d + \boldsymbol{b}^d$$



$$V(t,k) \neq V(t',k)$$

*ex) $V(0,k_1) \neq V(2,k_1)$

- **Evaluation function of the prism** for each state
- In upper stage, higher probability of actions leading to destination more efficiently

Prism-constrained RL model

Solve the value function by **backward induction**

- Initialize: $t = T$, $z_{T,d} = 1$ and $z_{t,k} = 0, \forall (t,k) \neq (T,d)$
- Set $t := t - 1$ and update the value function by

$$\boldsymbol{z}_t \leftarrow \mathbf{M}_t^{'} \boldsymbol{z}_{t+1} + \boldsymbol{b}$$

- If $t = 0$, finish the computation.

→ Efficiently solved and a unique solution is always found regardless of parameter value and magnitude of utility

*As long as T is finite,

$$\boldsymbol{z}_0 = \mathbf{M}_0^{'} \boldsymbol{z}_1 + \boldsymbol{b} = \mathbf{M}_0^{'}(\mathbf{M}_1^{'} \boldsymbol{z}_2 + \boldsymbol{b}) + \boldsymbol{b} = \cdots = (\mathbf{I} + \sum_{r=0}^{T-1} \prod_{s=0}^{r} \mathbf{M}_s^{'}) \boldsymbol{b}.$$

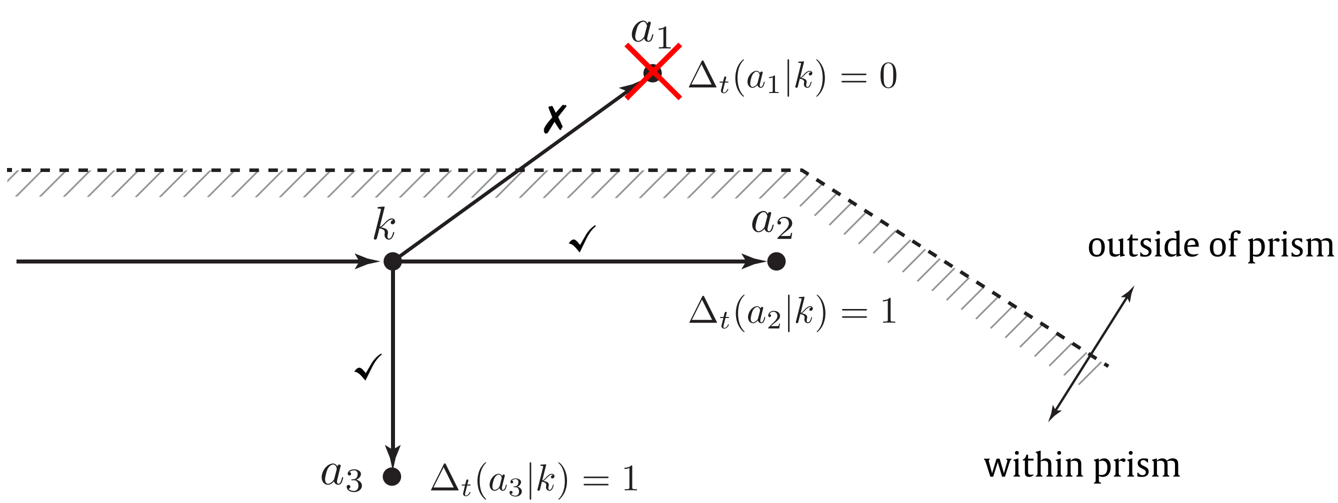
is theoretically upper bounded by a real vector because, for finite t,

$$\prod_{s=0}^{t} \mathbf{M}_s^{'} = \mathbf{M}_0^{'} \mathbf{M}_1^{'} \cdots \mathbf{M}_t^{'} \leq \mathbf{M}^t \leq C$$

Prism-constrained RL model

Action choice probability

$$p_t^d(a|k) = \frac{\Delta_t^d(a|k) e^{\frac{1}{\mu}\{v(a|k)+V^d(t+1,a)\}}}{\sum_{a' \in A(k)} \Delta_t^d(a'|k) e^{\frac{1}{\mu}\{v(a'|k)+V^d(t+1,a')\}}}$$



Maximum likelihood estimation

Translation of path observations

Original: $\sigma_n = [k_0, \ldots, k_{J_n}]$

Translated: $\sigma_n^* = [(0, k_0), \ldots, (J_n, d_n), (J_n + 1, d_n), \ldots (T, d_n)]$

Stay at destination after the arrival

(*Set T so that $T \geq \max_n J_n$)

Likelihood function

$$
\begin{aligned}
LL(\beta; \sigma^*) &\equiv \log \prod_{n=1}^{N} P(\sigma_n^*) \\
&= \sum_{n=1}^{N} \sum_{t=0}^{T-1} \log p_t^{d_n}(k_{t+1}|k_t) \\
&= \frac{1}{\mu} \sum_{n=1}^{N} \sum_{t=0}^{T-1} \left[v(k_{t+1}|k_t) + V^{d_n}(t+1, k_{t+1}) - V^{d_n}(t, k_t) \right]
\end{aligned}
$$

Numerical experiments

Questions

1. **Can the Prism-RL model reproduce the true parameter** of the RL model even when positive attributes are included ?
2. If yes, **how stable is the estimation w.r.t. starting points** ?

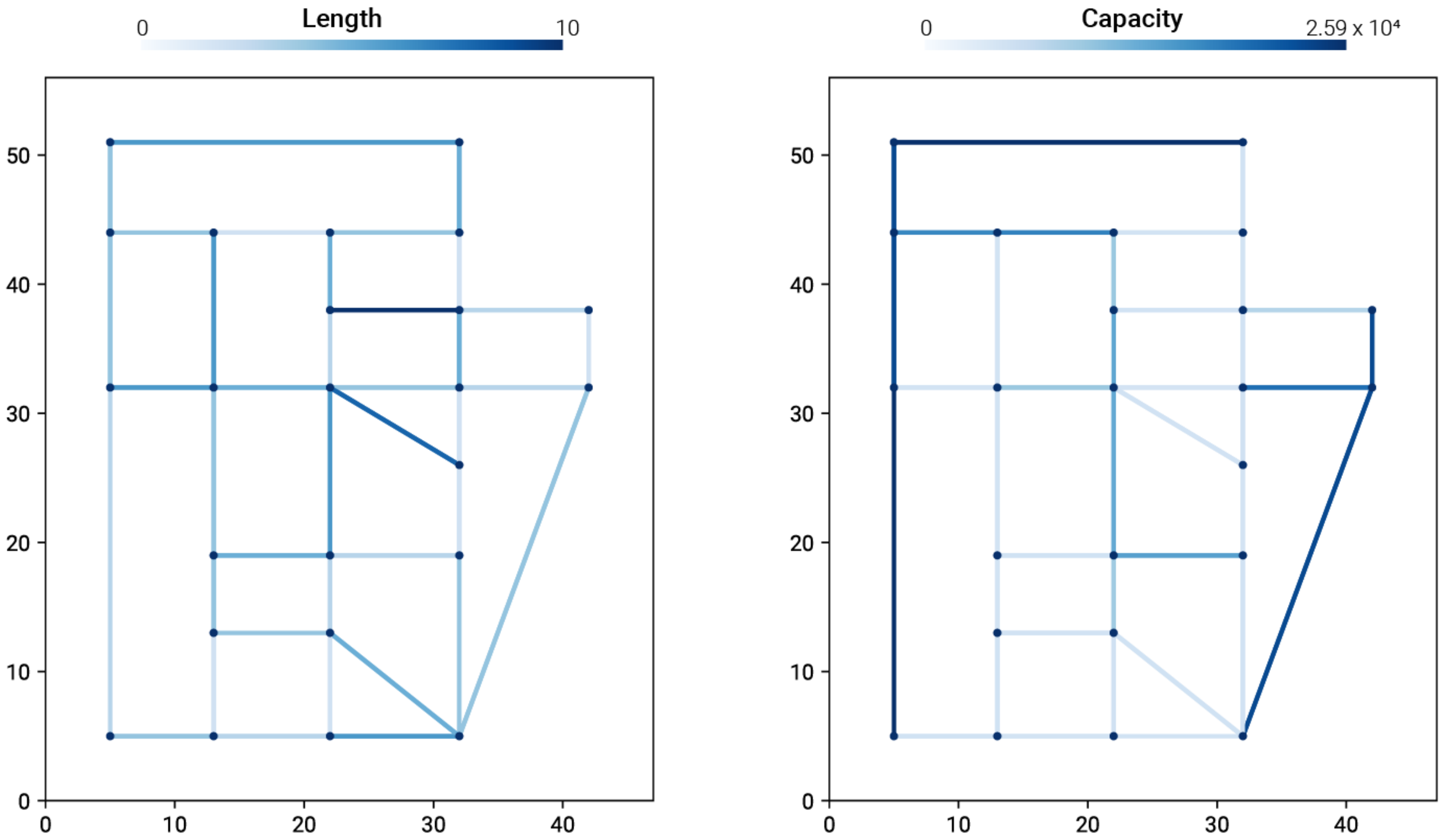
Experimental setup

- Sioux Falls network route choice
- Observations **simulated by the RL model with known true parameters** (24 OD x 1000 samples)
- Estimate both the RL and Prism-RL model **using the same observations**
- Prism-RL model
  - Define a state network for each destination
  - Set T = 15 (*did not affect the results)

Numerical experiments

$$v(a|k) = \beta_{\text{len}}\text{Length}_a + \beta_{\text{cap}}\text{Capacity}_a - 10\text{Uturn}_{a|k}$$

$$(\beta_{\text{len}} < 0; \beta_{\text{cap}} > 0)$$



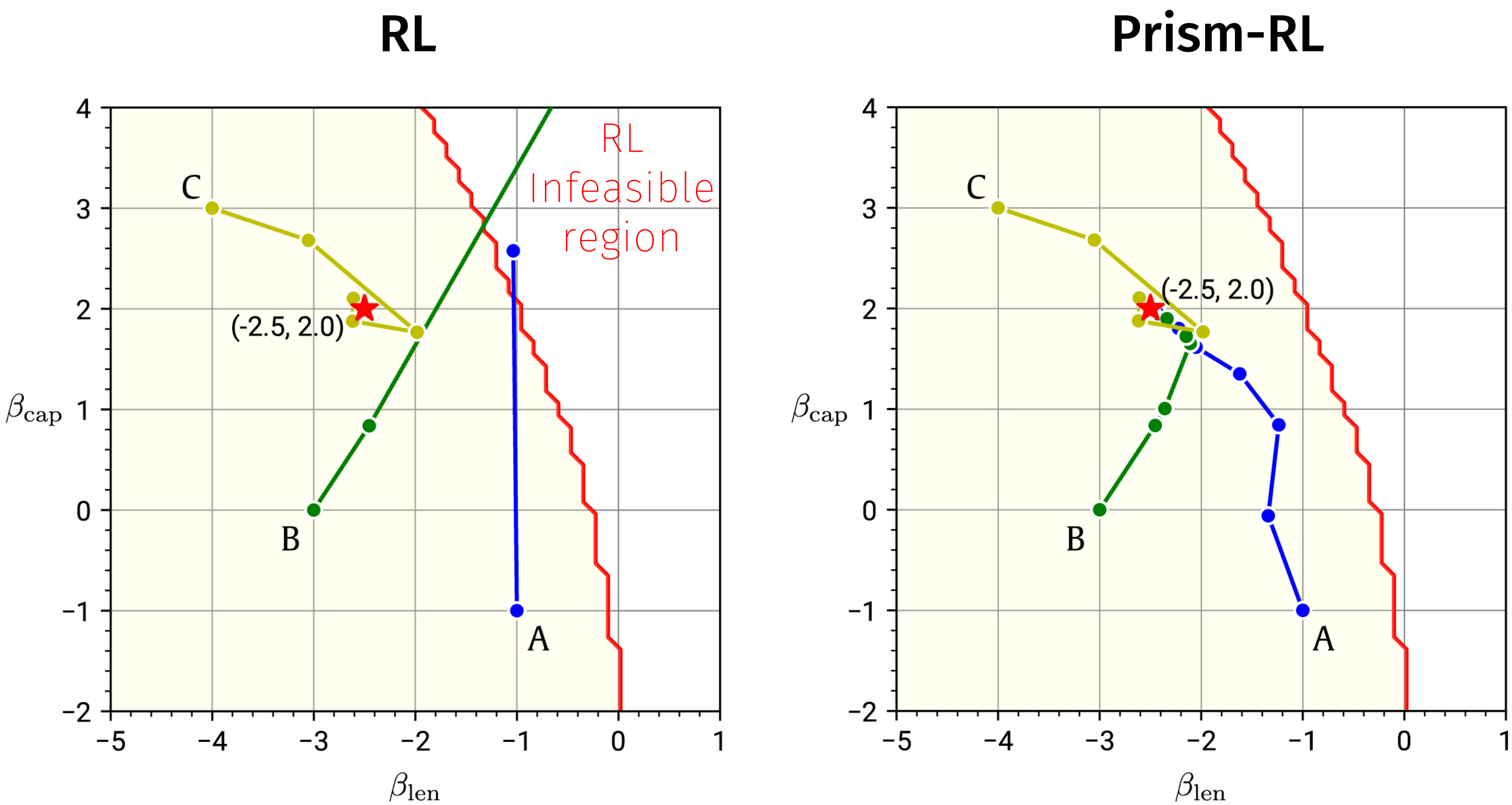
Numerical experiments | Estimation Result

β*	RL	Prism-RL
(-2.5, 2.0)	N/A	(-2.493**, 2.002**) 3.36s / LL=-4919.0
(-2.5, 1.5)	(-2.505**, 1.509**) 0.99s / LL=-5420.5	(-2.505**, 1.509**) 2.87s / LL=-5420.5
(-1.5, 1.5)	N/A	(-1.511**, 1.522**) 2.99s / LL=-7778.3
(-1.5, 1.0)	N/A	(-1.492**, 0.994**) 2.64s / LL=-8366.9

* All true params are feasible solution to the RL model; estimation started with (-1, -1)

- **Prism-RL model reproduced the true values with high accuracy** even for the cases where RL model failed
- In the case where RL was successfully estimated, the estimation results of both models were consistent

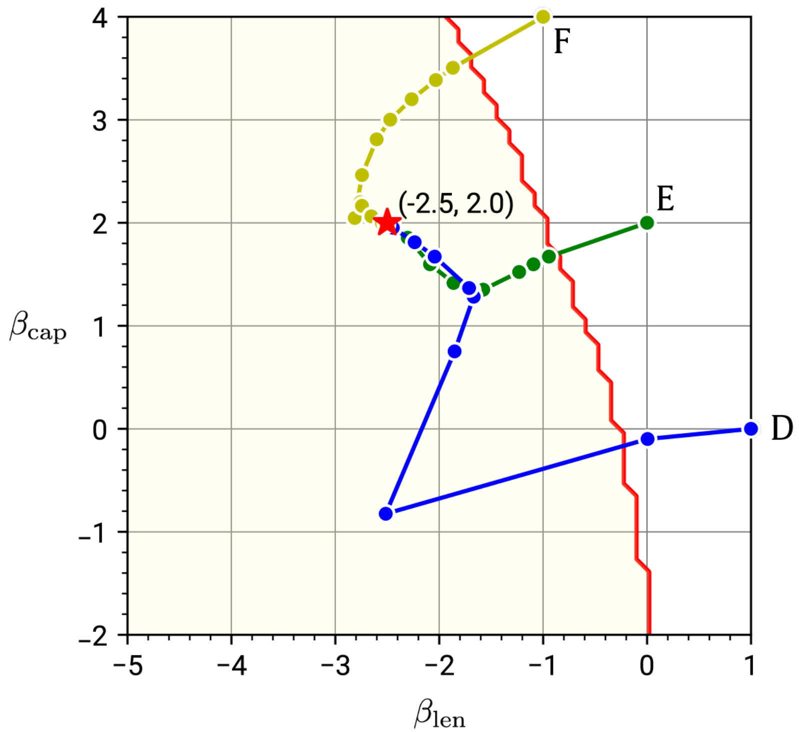
Numerical experiments | Estimation Process



- RL model depends on starting point and often diverge during the estimation
- **Prism-RL model converges to the true value regardless of starting point** (*update to infeasible region was not observed)

Numerical experiments | Estimation Process

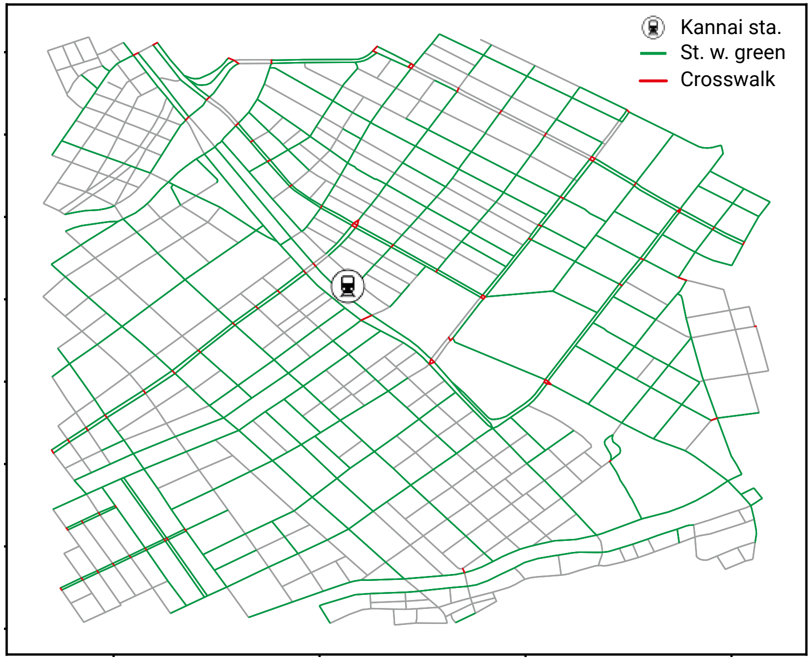
Prism-RL

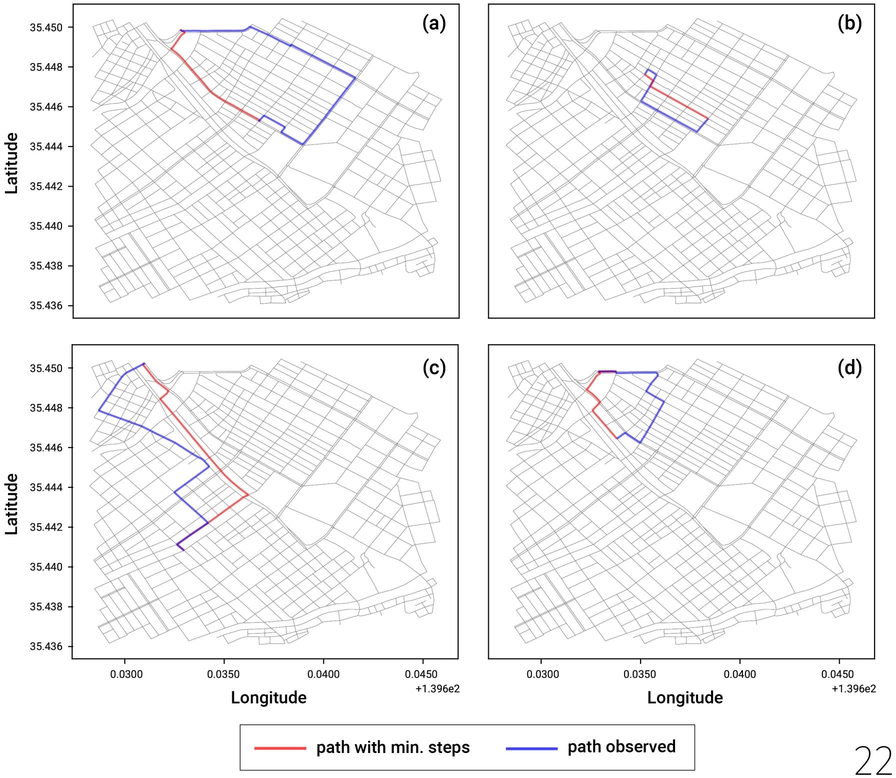


- Stably converged to the true parameter **even when starting point is outside the RL feasible region**
- No prior information on the true value is needed (you can set an initial point as you like)

Real application to pedestrian route choice

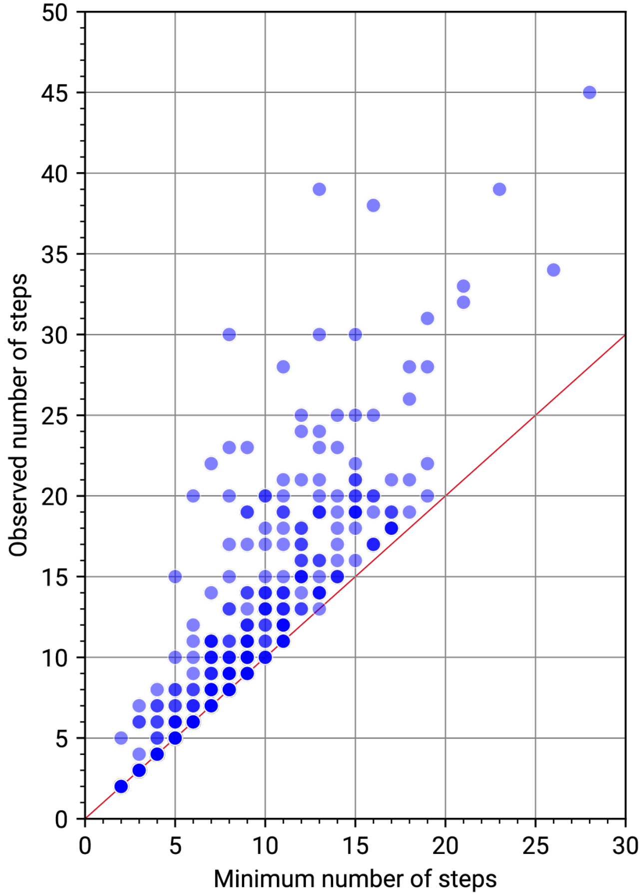
- **Kannai, Yokohama** (a mile square centered on Kannai station)
  - 724 Nodes, 2398 Links, 8434 Link pairs
- **PP Survey** in H30 PT survey
  - 410 observed paths of 159 pedestrians, 164 destinations
  - **Diverge walking paths** including large detours





Real application | Definition of choice-stage constraint T

count	mean	std	min	25%	50%	75%	max
410	1.24	0.39	1.00	1.00	1.08	1.33	3.75



- Define T for each d based on **observed detour rate**:

$$T_d \equiv \max_{n \in N_d} \{\max(1.34 \times D^d(o_n), J_n)\}$$

  - N_d : Set of observed paths for d
  - $D^d(o_n)$: min. steps b/w observed OD pair
  - J_n : observed no. of steps of n

- **75 percentile value** (=1.34) to include diverse paths in the path set
- All observations satisfy the prism constraint

Real application | Utility specification

- Compare two different utility specifications:

$$v(a|k) = \beta_{\text{len}}\text{Length}_a + \beta_{\text{cross}}\text{Crosswalk}_a - 10\text{Uturn}_{a|k}, \quad \text{(a)}$$

$$v(a|k) = (\beta_{\text{len}} + \underline{\beta_{\text{green}}\text{Green}_a})\text{Length}_a + \beta_{\text{cross}}\text{Crosswalk}_a - 10\text{Uturn}_{a|k}, \quad \text{(b)}$$

  - *Length*: length of link (m/10)
  - *Crosswalk*: 1 if the link is a crosswalk and 0 otherwise
  - *Green*: On-street green presence (1/0; interacted with link length)

To clarify:

1. Street greenery produces a positive effect on utility for pedestrian route choice? **Can Prism-RL model capture it**?
2. **How different are the estimation results** of RL/Prism-RL models?

Real application | Estimation result

	RL (a)	Prism-RL (a)	RL (b)	Prism-RL (b)
$\hat{\beta}_{len}$	**-0.297**	**-0.245**	-	**-0.266**
std.err.	0.008	0.007	-	0.020
t-test	-38.832	-37.264	-	-13.283
$\hat{\beta}_{cross}$	**-0.924**	**-0.774**	-	**-0.791**
std.err.	0.075	0.171	-	0.068
t-test	-12.237	-4.517	-	-11.638
$\hat{\beta}_{green}$			-	**0.049**
std.err.			-	0.010
t-test			-	4.817
LL	-1772.972	-1637.484	-	-1612.894
#paths	410	410	410	410

- For specification (a), **both RL and Prism-RL models obtained estimation results** with statistical significance
  - *Length*, *Crosswalk* are both negative → Pedestrians do not like paths with long distance and crosswalks
  - Prism-RL model fits better (details later)
- For (b), **ONLY Prism-RL model obtained the result** and **captured the positive utility** of green presence on streets
  - RL model failed with all the tested starting points
  - Adding the attribute improved goodness-of-fit (with 99% significance)

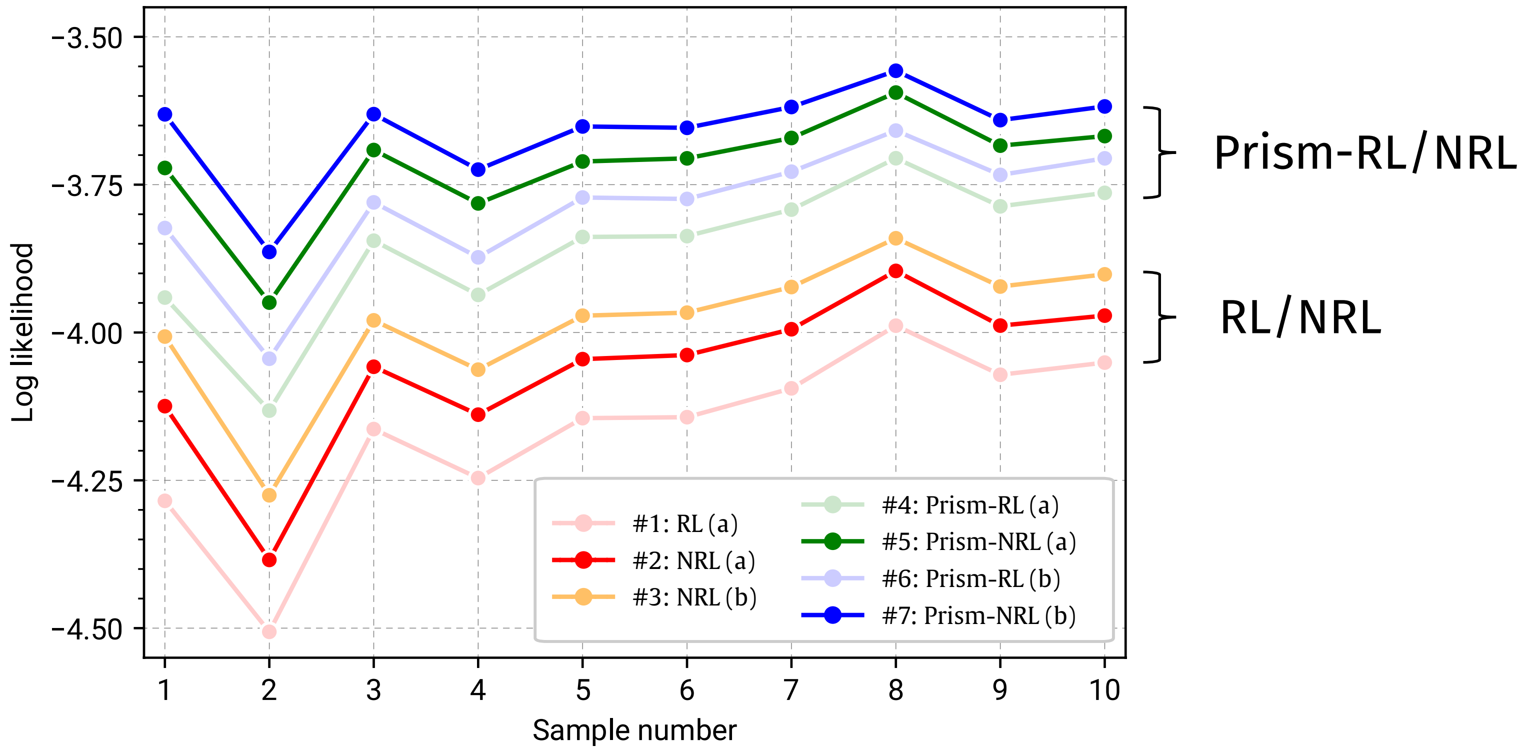
Real application | Estimation result - Nested models

	NRL (a)	Prism-NRL (a)	NRL (b)	Prism-NRL (b)
$\hat{\beta}_{\text{len}}$	**-0.460**	**-0.445**	**-0.485**	**-0.469**
std.err.	0.030	0.039	0.082	0.062
t-test	-15.166	-11.304	-5.945	-7.568
$\hat{\beta}_{\text{cross}}$	**-1.262**	**-1.206**	**-1.281**	**-1.206**
std.err.	0.163	0.120	0.280	0.201
t-test	-7.728	-10.021	-4.567	-5.993
$\hat{\beta}_{\text{green}}$	-	-	**0.078**	**0.082**
std.err.	-	-	0.021	0.014
t-test	-	-	3.690	5.855
$\hat{\omega}$	**0.064**	**0.095**	**0.063**	**0.091**
std.err.	0.006	0.013	0.012	0.010
t-test	9.942	7.402	5.459	8.769
LL	-1734.622	-1587.079	-1707.068	-1565.531
#paths	410	410	410	410

* $\mu_k^d = e^{\omega\sqrt{\text{SP}_{kd}}}$ where SP_{kd} is the shortest path length between k and d

- **Prism-based approach also suited the NRL model** as well
  - Same signs as RL models; Prism-NRL fits better than NRL
  - **Captured correlation: scale (variance) decreases on links close to destination**
- NRL could be estimated for spec. (b) but depended on starting point
  - **Prism-NRL was successfully estimated with all initial points** tested
  - Prism-based approach can also be viewed as **a good approximation to provide a nice starting point** for original RL models

Real application | Model validation



- Compare model performance of out-of-sample prediction
  - 10 sets of randomly split estimation and validation samples (8:2)
- <u>Prism-RL model shows a higher prediction performance for all samples</u>
  - Universal set (RL/NRL) vs Prism-based path set (Prism-RL/NRL)
  - Inclusion of positive attribute & nesting also improved the performance

Real application | Impact of T (Prism-RL results)

Test **different detour rates** γ in $T_d \equiv \max\left[\max_{n \in N_d}\{\gamma D^d(o_n), J_n\}\right]$

Small

Path set size

Large

γ	$\hat{\beta}_{\text{len}}$	$\hat{\beta}_{\text{cross}}$	$\hat{\beta}_{\text{green}}$	LL
1.25	-0.265	-0.785	0.049	-1605.104
1.34	-0.266	-0.791	0.049	-1612.894
1.50	-0.271	-0.796	0.050	-1632.277
2.00	-0.284	-0.817	0.052	-1661.690

*Corresponds to RL when *T* goes to infinity

- The signs and scales of the estimates **remained unchanged**
- The **ratio of negative parameters systematically increased** as γ grew
  - Prism-RL model adjusted the parameter to keep little probability of detour/cyclic paths
- Model **fits better with smaller γ values** (i.e., tighter constraints)
  - Due to the **exclusion of behaviorally unrealistic paths**
  - Trade-off with out-of-sample prediction (choice set consistency)

Conclusion

- Recursive model describes the **decision-making structure** and is a **computationally efficient method** in a **dynamic/sequential choice** context
- Recursive model entails a **numerical issue with respect to the value function** which is **critical during the estimation**
- Prism-based approach **solves the issue and stably captures positive network attributes**
- Expanded the practical applicability of recursive models

Thank you for your attention!

Cite/more details:
"Capturing positive network attributes during the estimation of recursive logit models: A prism-based approach"
arXiv:2204.01215 [econ.EM]

References

RL model and its numerical issues

- Fosgerau, M., Frejinger, E., Karlstrom, A. (2013). A link based network route choice model with unrestricted choice set. *Transportation Research Part B: Methodological* 56: 70–80.
- Oyama, Y., Hato, E., (2017) A discounted recursive logit model for dynamic gridlock network analysis. *Transportation Research Part C: Emerging Technologies* 85: 509–527.
- Mai, T., Frejinger, E. (2022). Undiscounted recursive path choice models: Convergence properties and algorithms. *Transportation Science*.

Prism-constrained RL model

- Oyama, Y. (2022) Capturing positive network attributes during the estimation of recursive logit models: A prism-based approach. arXiv: 2204.01215 [econ.EM].
- Oyama, Y., Hato, E. (2019) Prism-based path set restriction for solving Markovian traffic assignment problem. *Transportation Research Part B: Methodological* 122: 528-546.
- 大山雄己, 羽藤英二 (2017) 時間構造化ネットワーク上の確率的交通配分. 土木学会論文集 D3(土木計画学) 73(4): 186-200.
- 大山雄己, 羽藤英二 (2016) 時空間制約と経路相関を考慮した歩行者の活動配分問題. 都市計画論文集 51(3): 680-687.

Appendix | CPU time for estimation

Average CPU time (s) over 10 samples in validation

	RL	NRL		Prism-RL		Prism-NRL	
$v(a\|k)$	(a)	(a)	(b)	(a)	(b)	(a)	(b)
CPU time (s)	36.40	1096.82	1282.22	386.24	768.89	781.48	900.17

- RL model is very fast due to the linear system
- Advantage of prism-constrained models is **independence of the model linearity**: the scale of required CPU time does not change very much between Prism-RL and Prism-NRL models