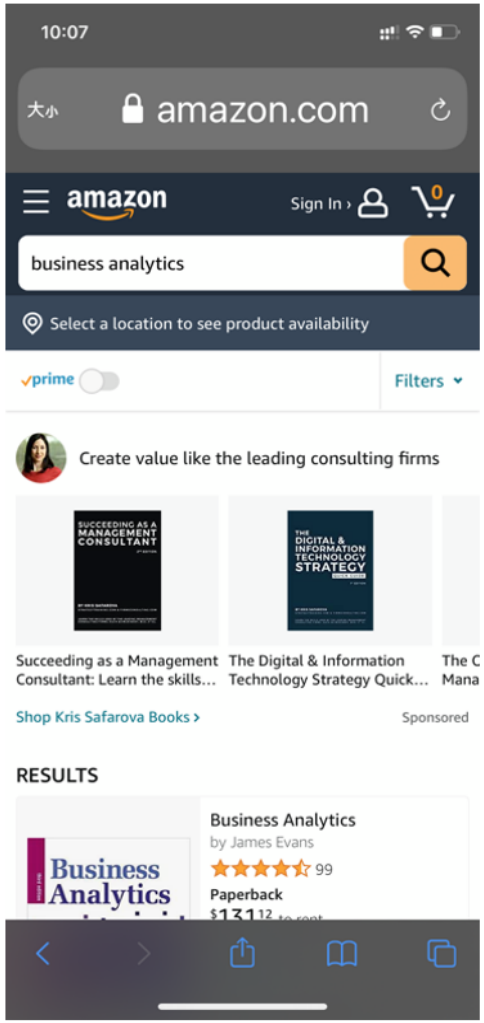
Safe Optimal Design with Applications in Off-Policy Learning

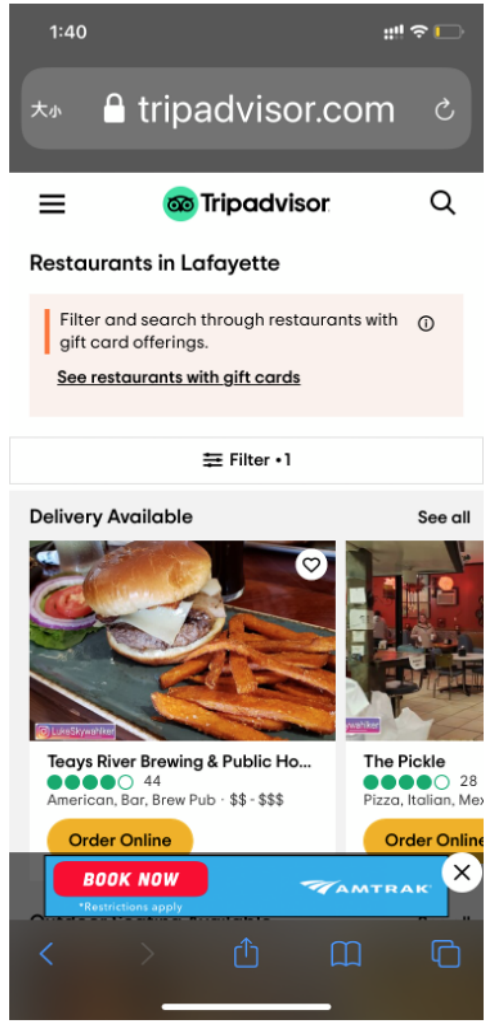
Ruihao Zhu

Purdue University Krannert School of Management
Email: rhzhu@purdue.edu

Joint work with Branislav Kveton (Amazon)

Recommendation Systems





The Decision-Maker

In online platforms, automated *policies* handle recommendation

- Direct user traffic to different items
- **Goal:** maximize reward, *e.g.*, #clicks, conversion rates

Challenges:

- Newly added items, users' interest shift ...

Therefore, platforms need to improve their policies continually

The Decision-Maker

In online platforms, automated *policies* handle recommendation

- Direct user traffic to different items
- **Goal:** maximize reward, *e.g.*, #clicks, conversion rates

Challenges:

- Newly added items, users' interest shift ...

Therefore, platforms need to improve their policies continually

The Decision-Maker

In online platforms, automated *policies* handle recommendation

- Direct user traffic to different items
- **Goal:** maximize reward, *e.g.*, #clicks, conversion rates

Challenges:

- Newly added items, users' interest shift ...

Therefore, platforms need to improve their policies continually

The Decision-Maker

In online platforms, automated *policies* handle recommendation

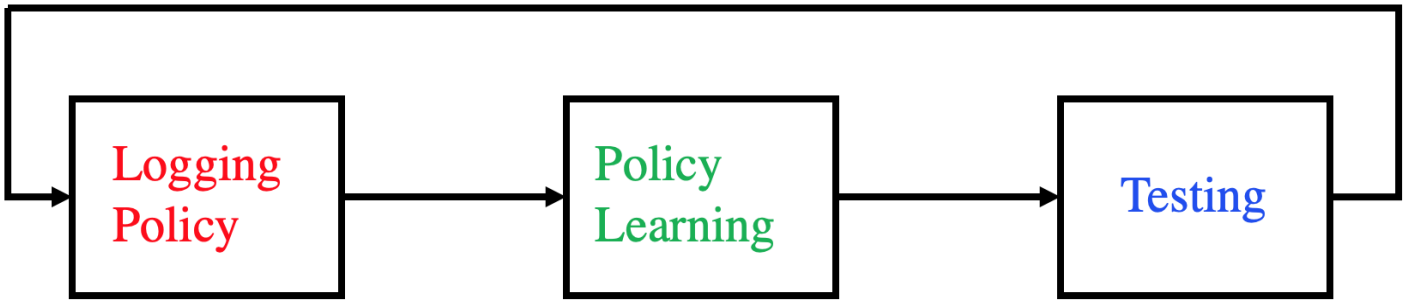
- Direct user traffic to different items
- **Goal:** maximize reward, *e.g.*, #clicks, conversion rates

Challenges:

- Newly added items, users' interest shift ...

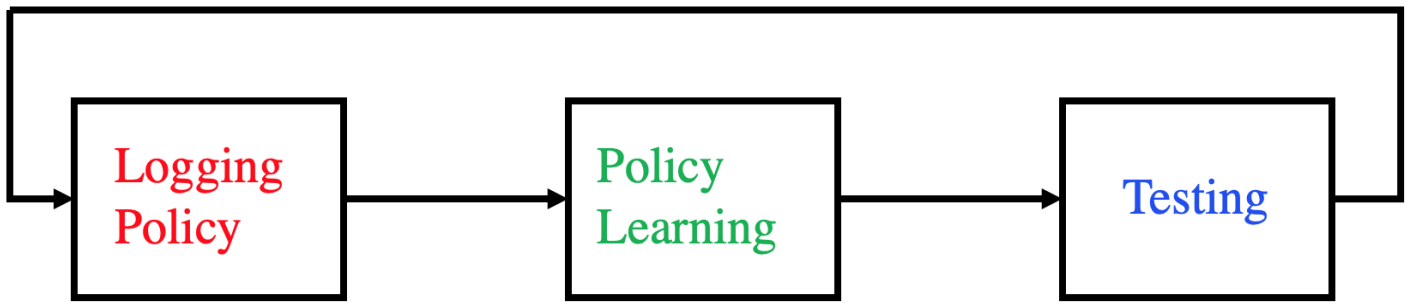
Therefore, platforms need to improve their policies continually

Developing Good Policies



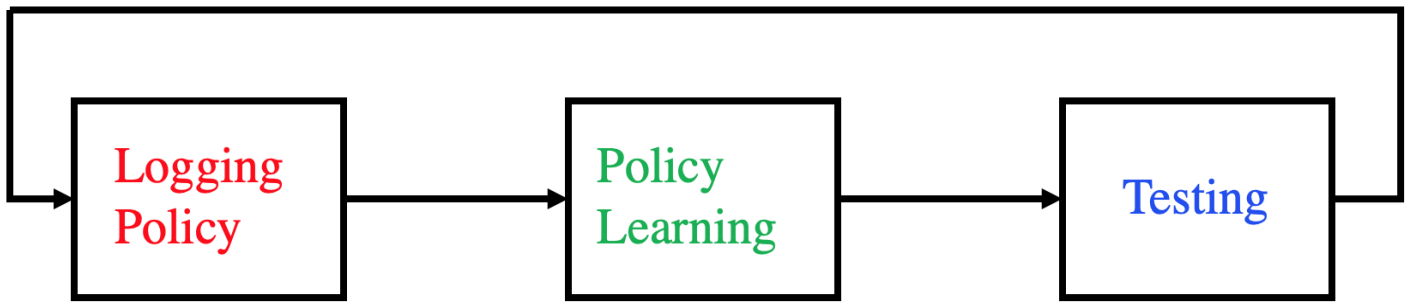
- **Logging Policy:** gain rewards and display different items to acquire data
- **Policy Learning:** use logged data to learn a new promising policy
- **Testing:** compare incumbent policy and the newly learned policy

Developing Good Policies



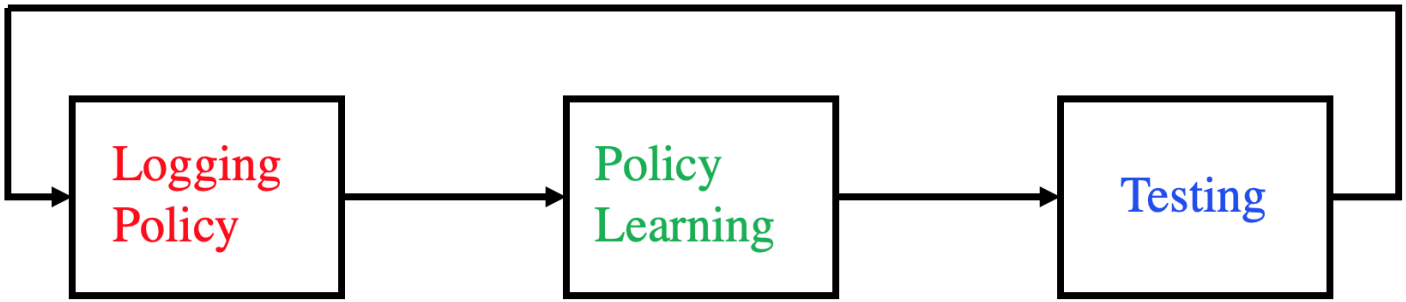
- **Logging Policy:** gain rewards and display different items to acquire data
- **Policy Learning:** use logged data to learn a new promising policy
- **Testing:** compare incumbent policy and the newly learned policy

Developing Good Policies



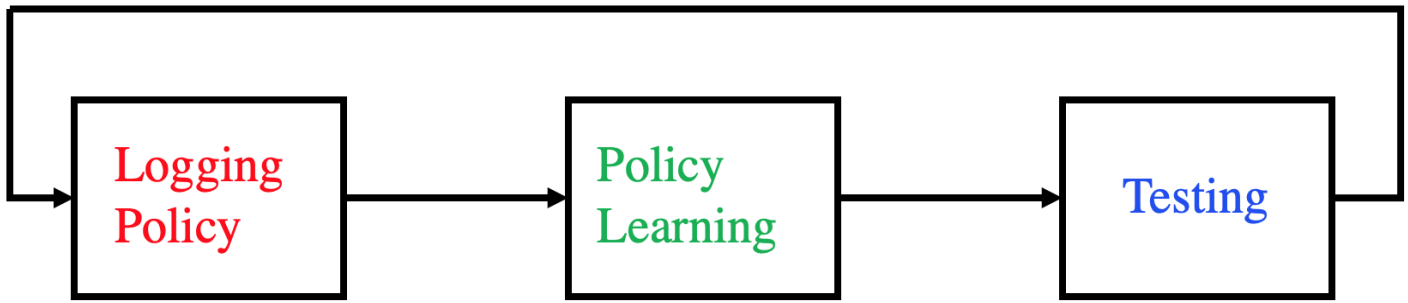
- **Logging Policy:** gain rewards and display different items to acquire data
- **Policy Learning:** use logged data to learn a new promising policy
- **Testing:** compare incumbent policy and the newly learned policy

Developing Good Policies



- **Logging Policy:** gain rewards and display different items to acquire data
- **Policy Learning:** use logged data to learn a new promising policy
- **Testing:** compare incumbent policy and the newly learned policy

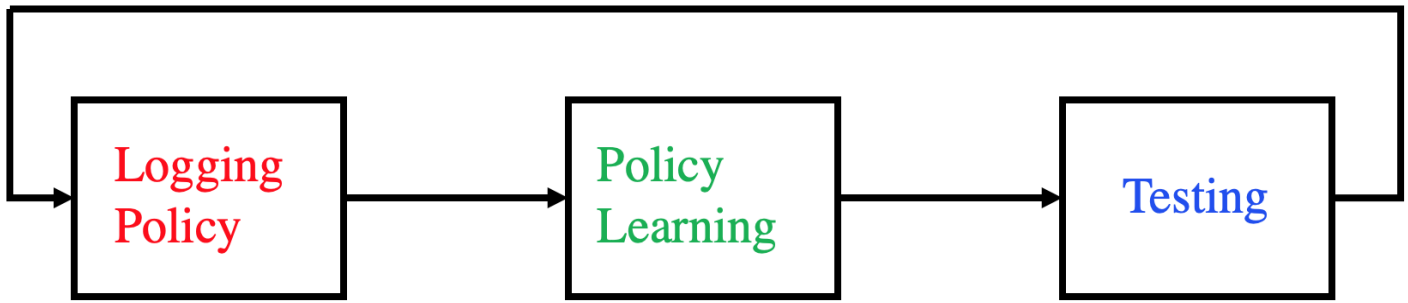
Developing Good Policies



Specifically:

- **Logging Policy:** modify the (baseline) production policy
- **Policy Learning:** carried out by different off-policy evaluation methods *(Li et al. 11, Dudik et al. 14)*
- **Testing:** A/B testing (against the baseline) to determine the winning policy (*i.e.*, the one with higher reward)

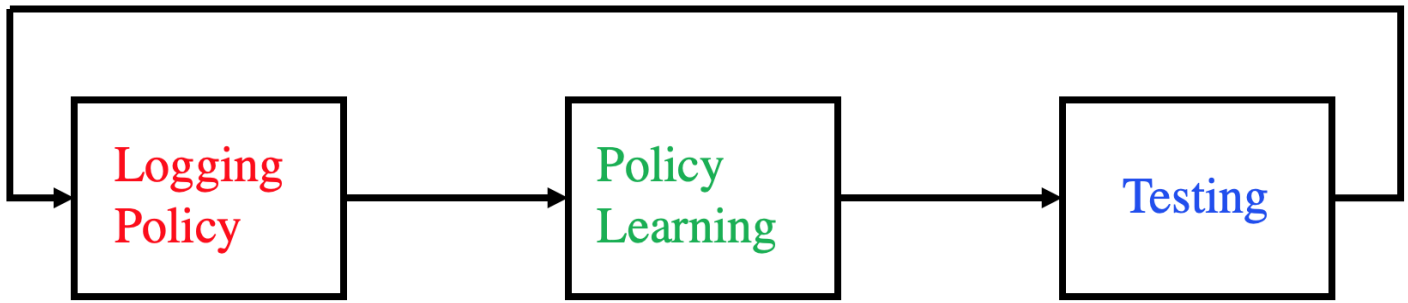
Developing Good Policies



Specifically:

- **Logging Policy:** modify the (baseline) production policy
- **Policy Learning:** carried out by different off-policy evaluation methods *(Li et al. 11, Dudik et al. 14)*
- **Testing:** A/B testing (against the baseline) to determine the winning policy (*i.e.*, the one with higher reward)

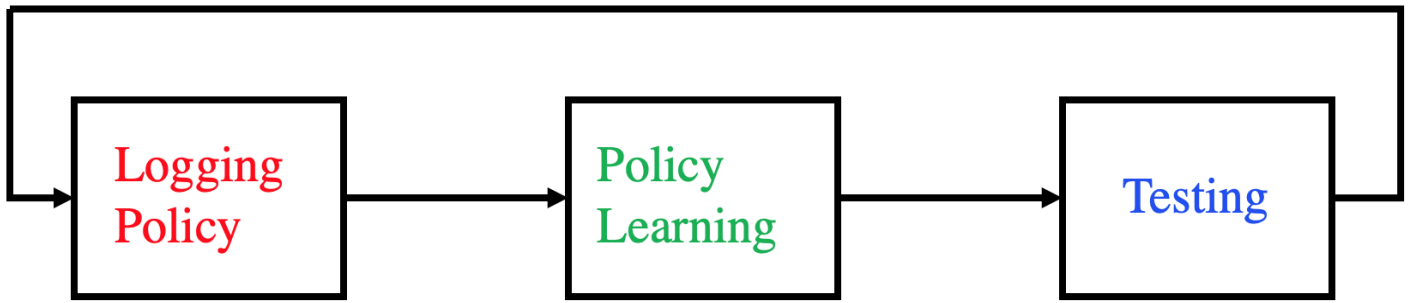
Developing Good Policies



Specifically:

- **Logging Policy:** modify the (baseline) production policy
- **Policy Learning:** carried out by different off-policy evaluation methods *(Li et al. 11, Dudik et al. 14)*
- **Testing:** A/B testing (against the baseline) to determine the winning policy (*i.e.*, the one with higher reward)

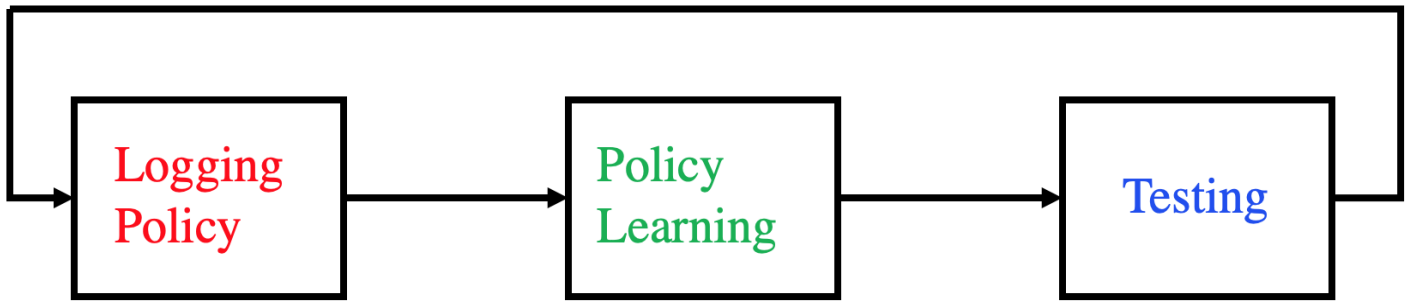
Developing Good Policies



Specifically:

- **Logging Policy:** modify the (baseline) production policy
- **Policy Learning:** carried out by different off-policy evaluation methods *(Li et al. 11, Dudik et al. 14)*
- **Testing:** A/B testing (against the baseline) to determine the winning policy (*i.e.*, the one with higher reward)

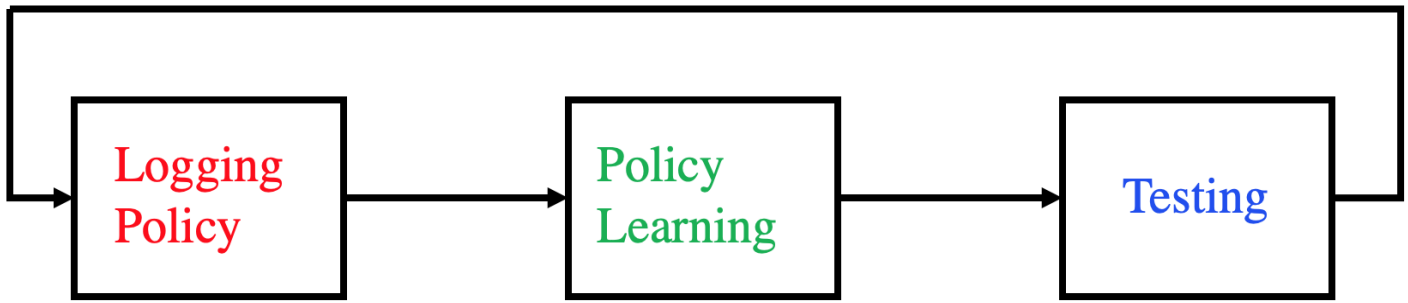
Developing Good Policies



Specifically:

- **Logging Policy:** modify the (baseline) production policy
- ✓ **Policy Learning:** carried out by different off-policy evaluation methods *(Li et al. 11, Dudik et al. 14)*
- ✓ **Testing:** A/B testing (against the baseline) to determine the winning policy (*i.e.*, the one with higher reward)

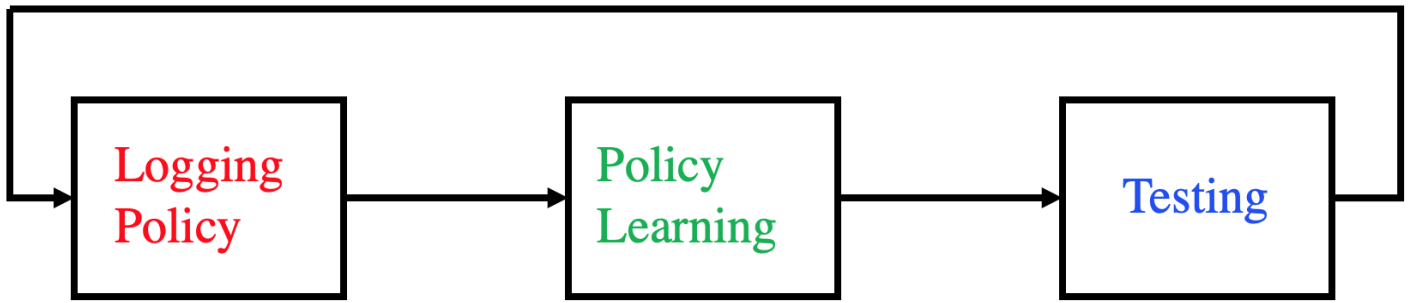
Developing Good Logging Policies



Successful policy learning relies on sufficiently explored logged data *(Wang et al. 17)*

- Ideally, logging policy would try to allocate equal traffic to each item to *gain information*
- But logging policy also needs to satisfy some *safety constraint* to avoid costly data collection

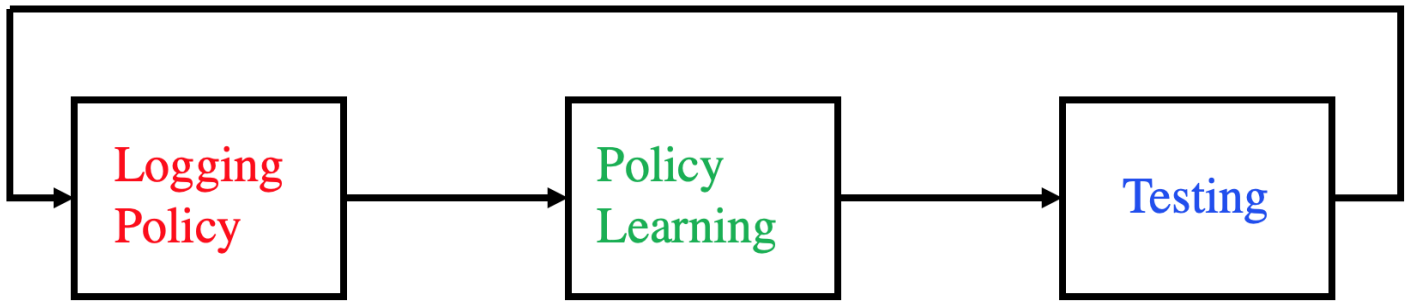
Developing Good Logging Policies



Successful policy learning relies on sufficiently explored logged data *(Wang et al. 17)*

- Ideally, logging policy would try to allocate equal traffic to each item to *gain information*
- But logging policy also needs to satisfy some *safety constraint* to avoid costly data collection

Developing Good Logging Policies



Successful policy learning relies on sufficiently explored logged data *(Wang et al. 17)*

- Ideally, logging policy would try to allocate equal traffic to each item to *gain information*
- But logging policy also needs to satisfy some *safety constraint* to avoid costly data collection

Results Overview

- Current practice (production policy + uniform exploration), despite being safe, is sub-optimal in general
  - **Metric:** minimize maximum variance of estimation error (G-optimal design)
- Safe optimal logging policy in the worst case
  - **Water-Filling Method:** incrementally allocates user traffic to the item that needs exploration the most
- Incorporate side information (*e.g.*, historical data) and extend to the linear contextual model
- Implications for downstream policy learning through theoretical analysis and extensive numerical experiments

Results Overview

- Current practice (production policy + uniform exploration), despite being safe, is sub-optimal in general
  - **Metric:** minimize maximum variance of estimation error (G-optimal design)
- Safe optimal logging policy in the worst case
  - **Water-Filling Method:** incrementally allocates user traffic to the item that needs exploration the most
- Incorporate side information (*e.g.*, historical data) and extend to the linear contextual model
- Implications for downstream policy learning through theoretical analysis and extensive numerical experiments

Results Overview

- Current practice (production policy + uniform exploration), despite being safe, is sub-optimal in general
  - **Metric:** minimize maximum variance of estimation error (G-optimal design)
- Safe optimal logging policy in the worst case
  - **Water-Filling Method:** incrementally allocates user traffic to the item that needs exploration the most
- Incorporate side information (*e.g.*, historical data) and extend to the linear contextual model
- Implications for downstream policy learning through theoretical analysis and extensive numerical experiments