Contextual Bandits & a Real-world RL Case Study

Lucas Janson and Sham Kakade

CS/Stat 184: Introduction to Reinforcement Learning
Fall 2023

Today

- Contextual Bandits
- LinUCB
- Real world RL example

Contextual bandit environment

Formally, a contextual bandit is the following interactive learning process:

For $t = 0 \to T - 1$

1. Learner sees context $x_t \sim \nu_x$ — Independent of any previous data
2. Learner pulls arm $a_t = \pi_t(x_t) \in \{1, \ldots, K\}$ — π_t policy learned from all data seen so far
3. Learner observes reward $r_t \sim \nu^{(a_t)}(x_t)$ from arm a_t in context x_t

Note that if the context distribution ν_x always returns the same value (e.g., 0), then the contextual bandit reduces to the original multi-armed bandit

UCB for contextual bandits

UCB for contextual bandits

UCB algorithm conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

UCB for contextual bandits

UCB algorithm conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context

UCB for contextual bandits

UCB algorithm conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

UCB for contextual bandits

UCB algorithm conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

But when $|\mathcal{X}|$ is really big (or even infinite), this will be really bad!

UCB for contextual bandits

UCB algorithm conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

But when $|\mathcal{X}|$ is really big (or even infinite), this will be really bad!

Solution: share information across contexts x_t, i.e., don't treat $\nu^{(k)}(x)$ and $\nu^{(k)}(x')$ as completely different distributions which have nothing to do with one another

UCB for contextual bandits

UCB algorithm conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

But when $|\mathcal{X}|$ is really big (or even infinite), this will be really bad!

<ins>Solution</ins>: share information across contexts x_t, i.e., <ins>don't</ins> treat $\nu^{(k)}(x)$ and $\nu^{(k)}(x')$ as completely different distributions which have nothing to do with one another

<ins>Example</ins>: showing an ad on a NYT article on politics vs a NYT article on sports:

UCB for contextual bandits

UCB algorithm conceptually identical as long as $|\mathcal{X}|$ finite:

$$\pi_t(x_t) = \arg\max_k \hat{\mu}_t^{(k)}(x_t) + \sqrt{\ln(2TK|\mathcal{X}|/\delta)/2N_t^{(k)}(x_t)}$$

- Added x_t argument to $\hat{\mu}_t^{(k)}$ and $N_t^{(k)}$ since we now keep track of the sample mean and number of arm pulls *separately* for each value of the context
- Added $|\mathcal{X}|$ inside the log because our union bound argument is now over all arm mean estimates $\hat{\mu}_t^{(k)}(x)$, of which there are $K|\mathcal{X}|$ instead of just K

But when $|\mathcal{X}|$ is really big (or even infinite), this will be really bad!

Solution: share information across contexts x_t, i.e., don't treat $\nu^{(k)}(x)$ and $\nu^{(k)}(x')$ as completely different distributions which have nothing to do with one another

Example: showing an ad on a NYT article on politics vs a NYT article on sports: Not *identical* readership, but still both on NYT, so probably still *similar* readership!

Today

- Contextual Bandits
- LinUCB
- Real world RL example

Modeling in contextual bandits

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

$|\mathcal{X}| = 4$ $\Rightarrow$ w/o linear model, need to learn 4 different $\mu^{(k)}(x)$ values for each arm k

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

$|\mathcal{X}| = 4$ $\Rightarrow$ w/o linear model, need to learn 4 different $\mu^{(k)}(x)$ values for each arm k

With linear model there are just 2 parameters: the two entries of $\theta_k \in \mathbb{R}^2$

Modeling in contextual bandits

Need a model for $\mu^{(k)}(x)$, e.g., a linear model: $\mu^{(k)}(x) = \theta_k^\top x$

E.g., placing ads on NYT or WSJ (encoded as 0 or 1 in the first entry of x), for articles on politics or sports (encoded as 0 or 1 in the second entry of x) $\Rightarrow x \in \{0,1\}^2$

$|\mathcal{X}| = 4$ $\Rightarrow$ w/o linear model, need to learn 4 different $\mu^{(k)}(x)$ values for each arm k

With linear model there are just 2 parameters: the two entries of $\theta_k \in \mathbb{R}^2$

Lower dimension makes learning easier, but model could be wrong/biased

Linear model fitting

Linear model for rewards: $\mu^{(k)}(x) = x^\top \theta^{(k)}$

Linear model fitting

Linear model for rewards: $\mu^{(k)}(x) = x^\top \theta^{(k)}$

Least squares estimator: $\hat{\theta}_t^{(k)} = \arg\min_{\theta \in \mathbb{R}^d} \sum_{\tau=0}^{t-1} (r_\tau - x_\tau^\top \theta)^2 1_{\{a_\tau = k\}}$

Minimize squared error over time points when arm k selected

Linear model fitting

Linear model for rewards: $\mu^{(k)}(x) = x^\top \theta^{(k)}$

Least squares estimator: $\hat{\theta}_t^{(k)} = \arg\min_{\theta \in \mathbb{R}^d} \sum_{\tau=0}^{t-1} (r_\tau - x_\tau^\top \theta)^2 1_{\{a_\tau = k\}}$

Minimize squared error over time points when arm k selected

$$\hat{\theta}_t^{(k)} = \left(\sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau=k\}} \right)^{-1} \sum_{\tau=0}^{t-1} x_\tau r_\tau 1_{\{a_\tau=k\}}$$

$(X^\top X)^{-1}$ $X^\top Y$

Uncertainty quantification

Uncertainty quantification

For UCB, recall that we need confidence bounds on the expected reward of each arm (given context x_t)

Uncertainty quantification

For UCB, recall that we need confidence bounds on the expected reward of each arm (given context x_t)

Hoeffding was the main tool so far, but it used the fact that our estimate for the expected reward was a sample mean of the rewards we'd seen so far in the same setting (action, context)

Uncertainty quantification

For UCB, recall that we need confidence bounds on the expected reward of each arm (given context x_t)

Hoeffding was the main tool so far, but it used the fact that our estimate for the expected reward was a sample mean of the rewards we'd seen so far in the same setting (action, context)

With a model, we can use rewards we've seen in other settings → better estimation

Uncertainty quantification

For UCB, recall that we need confidence bounds on the expected reward of each arm (given context x_t)

Hoeffding was the main tool so far, but it used the fact that our estimate for the expected reward was a sample mean of the rewards we'd seen so far in the same setting (action, context)

With a model, we can use rewards we've seen in other settings → better estimation

But not using sample mean as estimator, so need something other than Hoeffding

Uncertainty quantification

For UCB, recall that we need confidence bounds on the expected reward of each arm (given context x_t)

Hoeffding was the main tool so far, but it used the fact that our estimate for the expected reward was a sample mean of the rewards we'd seen so far in the same setting (action, context)

With a model, we can use rewards we've seen in other settings → better estimation

But not using sample mean as estimator, so need something other than Hoeffding

Chebyshev's inequality: for a mean-zero random variable Y,

$|Y| \leq \beta\sqrt{\mathbb{E}[Y^2]}$ with probability $\geq 1 - 1/\beta^2$

Uncertainty quantification

For UCB, recall that we need confidence bounds on the expected reward of each arm (given context x_t)

Hoeffding was the main tool so far, but it used the fact that our estimate for the expected reward was a sample mean of the rewards we'd seen so far in the same setting (action, context)

With a model, we can use rewards we've seen in other settings → better estimation

But not using sample mean as estimator, so need something other than Hoeffding

Chebyshev's inequality: for a mean-zero random variable Y,

$$|Y| \leq \beta\sqrt{\mathbb{E}[Y^2]} \quad \text{with probability } \geq 1 - 1/\beta^2$$

Apply to $x_t^\top \hat{\theta}_t^{(k)} - x_t^\top \theta^{(k)}$

Chebyshev confidence bounds + intuition

Chebyshev confidence bounds + intuition

Chebyshev: $x_t^\top \theta^{(k)} \leq x_t^\top \hat{\theta}_t^{(k)} + \beta \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t}$ with probability $\geq 1 - 1/\beta^2$

$$A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}}$$

Chebyshev confidence bounds + intuition

Chebyshev: $x_t^\top \theta^{(k)} \leq x_t^\top \hat{\theta}_t^{(k)} + \beta \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t}$ with probability $\geq 1 - 1/\beta^2$

$$A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}}$$

Intuition:

Chebyshev confidence bounds + intuition

Chebyshev: $x_t^\top \theta^{(k)} \leq x_t^\top \hat{\theta}_t^{(k)} + \beta \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t}$ with probability $\geq 1 - 1/\beta^2$

$$A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}}$$

Intuition:

UCB term 1: $x_t^\top \hat{\theta}^{(k)}$ large when context and coefficient estimate aligned

Chebyshev confidence bounds + intuition

Chebyshev: $x_t^\top \theta^{(k)} \leq x_t^\top \hat{\theta}_t^{(k)} + \beta \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t}$ with probability $\geq 1 - 1/\beta^2$

$$A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}}$$

Intuition:

UCB term 1: $x_t^\top \hat{\theta}^{(k)}$ large when context and coefficient estimate aligned

UCB term 2: $x_t^\top (A_t^{(k)})^{-1} x_t = \frac{1}{N_t^{(k)}} x_t^\top (\Sigma_t^{(k)})^{-1} x_t$, where

$\Sigma_t^{(k)} = \frac{1}{N_t^{(k)}} A_t^{(k)} = \frac{1}{N_t^{(k)}} \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}}$ is the empirical covariance matrix of contexts when arm k chosen

Chebyshev confidence bounds + intuition

Chebyshev: $x_t^\top \theta^{(k)} \leq x_t^\top \hat{\theta}_t^{(k)} + \beta \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t}$ with probability $\geq 1 - 1/\beta^2$

$$A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}}$$

Intuition:

UCB term 1: $x_t^\top \hat{\theta}^{(k)}$ large when context and coefficient estimate aligned

UCB term 2: $x_t^\top (A_t^{(k)})^{-1} x_t = \frac{1}{N_t^{(k)}} x_t^\top (\Sigma_t^{(k)})^{-1} x_t$, where

$\Sigma_t^{(k)} = \frac{1}{N_t^{(k)}} A_t^{(k)} = \frac{1}{N_t^{(k)}} \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}}$ is the empirical covariance matrix of contexts when arm k chosen

Large when $N_t^{(k)}$ small or x_t not aligned with historical data

LinUCB algorithm

For $t = 0 \rightarrow T - 1$

LinUCB algorithm

For $t = 0 \rightarrow T - 1$

1. $\forall k$, define $A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}} + \lambda I$ and $\hat{\theta}_t^{(k)} = (A_t^{(k)})^{-1} \sum_{\tau=0}^{t-1} x_\tau r_\tau 1_{\{a_\tau = k\}}$

LinUCB algorithm

Regularization makes $A_t^{(k)}$ invertible

For $t = 0 \to T-1$

1. $\forall k$, define $A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}} + \lambda I$ and $\hat{\theta}_t^{(k)} = (A_t^{(k)})^{-1} \sum_{\tau=0}^{t-1} x_\tau r_\tau 1_{\{a_\tau = k\}}$

LinUCB algorithm

Regularization makes $A_t^{(k)}$ invertible

For $t = 0 \to T - 1$

1. $\forall\, k$, define $A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}} + \lambda I$ and $\hat{\theta}_t^{(k)} = (A_t^{(k)})^{-1} \sum_{\tau=0}^{t-1} x_\tau r_\tau 1_{\{a_\tau = k\}}$

2. Observe context x_t and choose $a_t = \arg\max_k \left\{ x_t^\top \hat{\theta}_t^{(k)} + c_t \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t} \right\}$

LinUCB algorithm

Regularization makes $A_t^{(k)}$ invertible

For $t = 0 \to T-1$

1. $\forall k$, define $A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}} + \lambda I$ and $\hat{\theta}_t^{(k)} = (A_t^{(k)})^{-1} \sum_{\tau=0}^{t-1} x_\tau r_\tau 1_{\{a_\tau = k\}}$

2. Observe context x_t and choose $a_t = \arg\max_k \left\{ x_t^\top \hat{\theta}_t^{(k)} + c_t \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t} \right\}$

3. Observe reward $r_t \sim \nu^{(a_t)}(x_t)$

LinUCB algorithm

Regularization makes $A_t^{(k)}$ invertible

For $t = 0 \to T-1$

1. $\forall\, k$, define $A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}} + \lambda I$ and $\hat{\theta}_t^{(k)} = (A_t^{(k)})^{-1} \sum_{\tau=0}^{t-1} x_\tau r_\tau 1_{\{a_\tau = k\}}$
2. Observe context x_t and choose $a_t = \arg\max_k \left\{ x_t^\top \hat{\theta}_t^{(k)} + c_t \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t} \right\}$
3. Observe reward $r_t \sim \nu^{(a_t)}(x_t)$

c_t similar to log term in (non-lin)UCB, in that it depends logarithmically on

i. $1/\delta$ (δ is probability you want the bound to hold with)

ii. t and d implicitly via $\det(A_t^{(k)})$

LinUCB algorithm

Regularization makes $A_t^{(k)}$ invertible

For $t = 0 \rightarrow T-1$

1. $\forall\, k$, define $A_t^{(k)} = \sum_{\tau=0}^{t-1} x_\tau x_\tau^\top 1_{\{a_\tau = k\}} + \lambda I$ and $\hat{\theta}_t^{(k)} = (A_t^{(k)})^{-1} \sum_{\tau=0}^{t-1} x_\tau r_\tau 1_{\{a_\tau = k\}}$

2. Observe context x_t and choose $a_t = \arg\max_k \left\{ x_t^\top \hat{\theta}_t^{(k)} + c_t \sqrt{x_t^\top (A_t^{(k)})^{-1} x_t} \right\}$

3. Observe reward $r_t \sim \nu^{(a_t)}(x_t)$

c_t similar to log term in (non-lin)UCB, in that it depends logarithmically on

i. $1/\delta$ (δ is probability you want the bound to hold with)

ii. t and d implicitly via $\det(A_t^{(k)})$

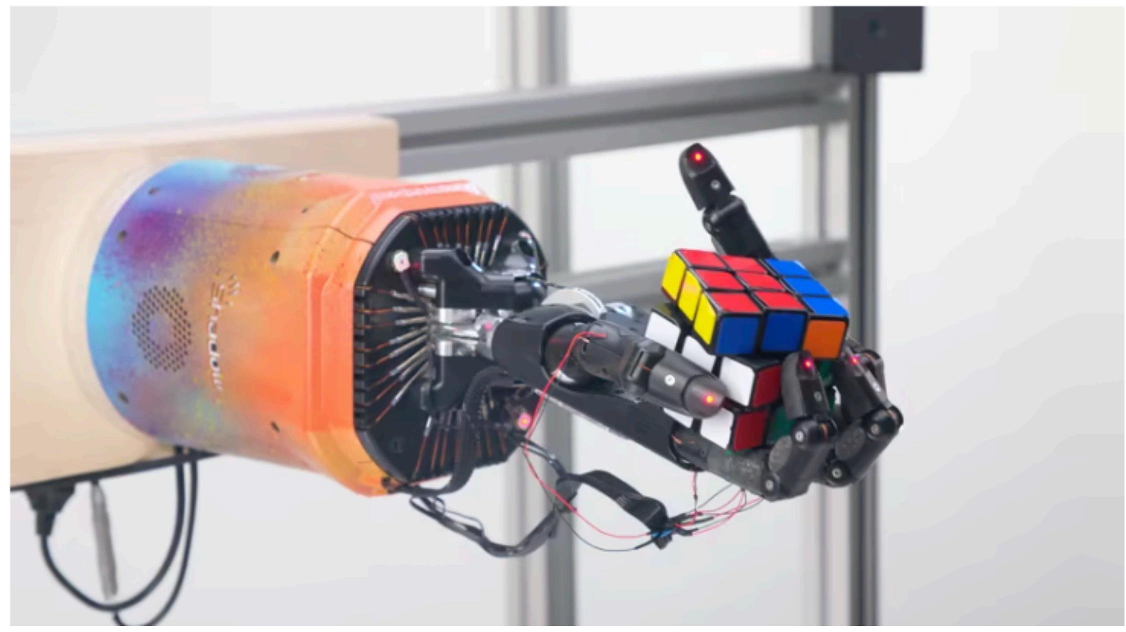
Can prove $\tilde{O}(\sqrt{T})$ regret bound

Today

- ✓ Contextual Bandits
- ✓ LinUCB
- ✓ Real world RL example

Case Study: RL for Supply Chains

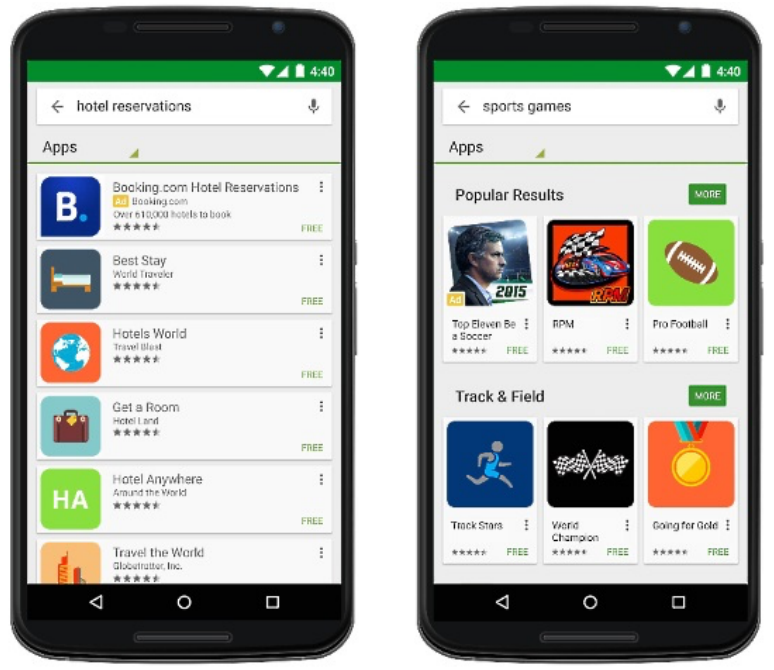
Real-world RL is hard.



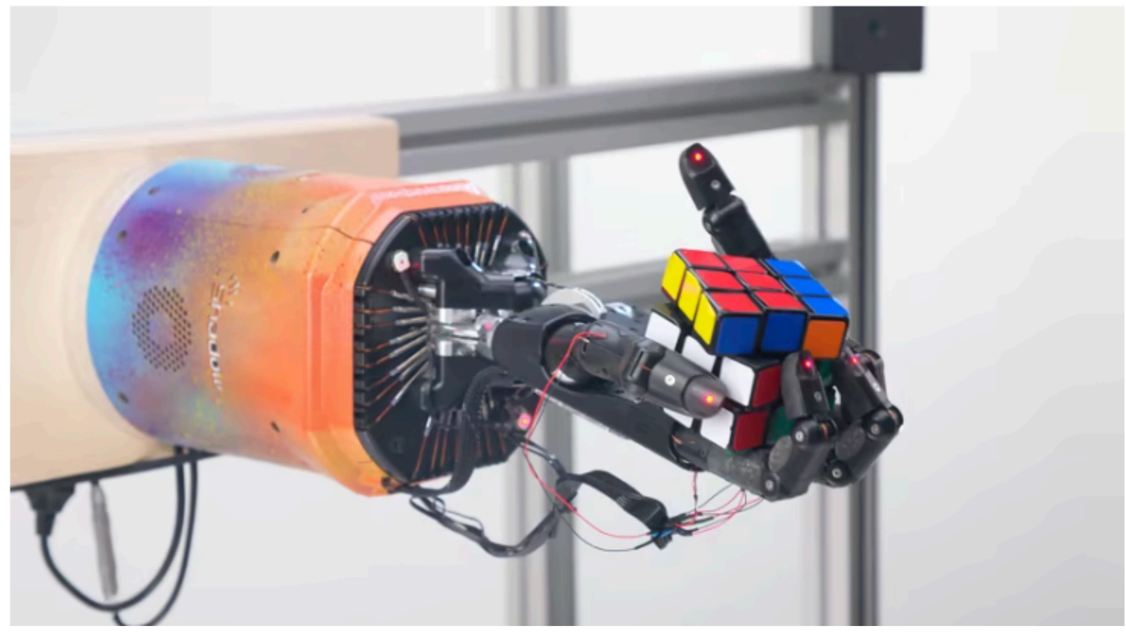
Many RL successes in controlled domains.

How can RL add value in the real world?

ChatGPT



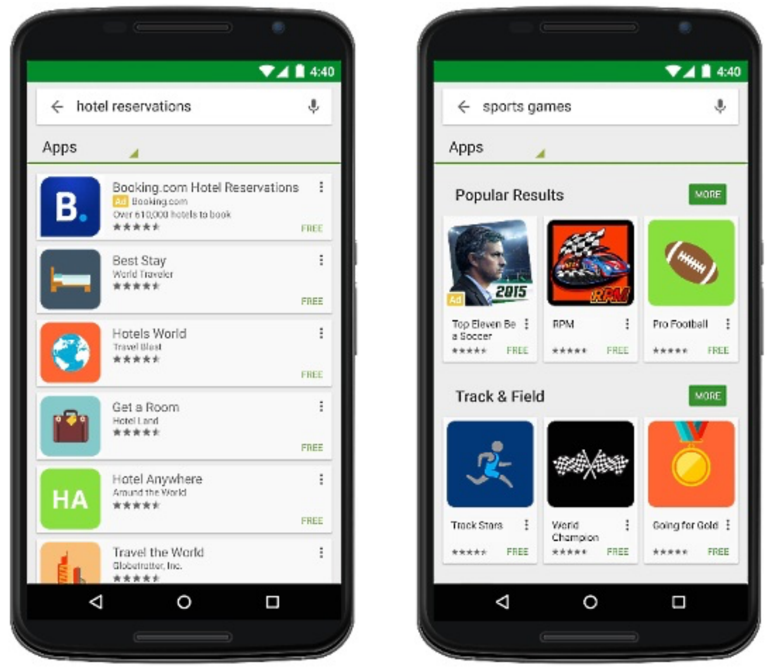
Real-world RL is hard.



Many RL successes in controlled domains.

How can RL add value in the real world?

ChatGPT



Issues:
sample complexity?
how to use offline data?
exploration/counterfactual reasoning?

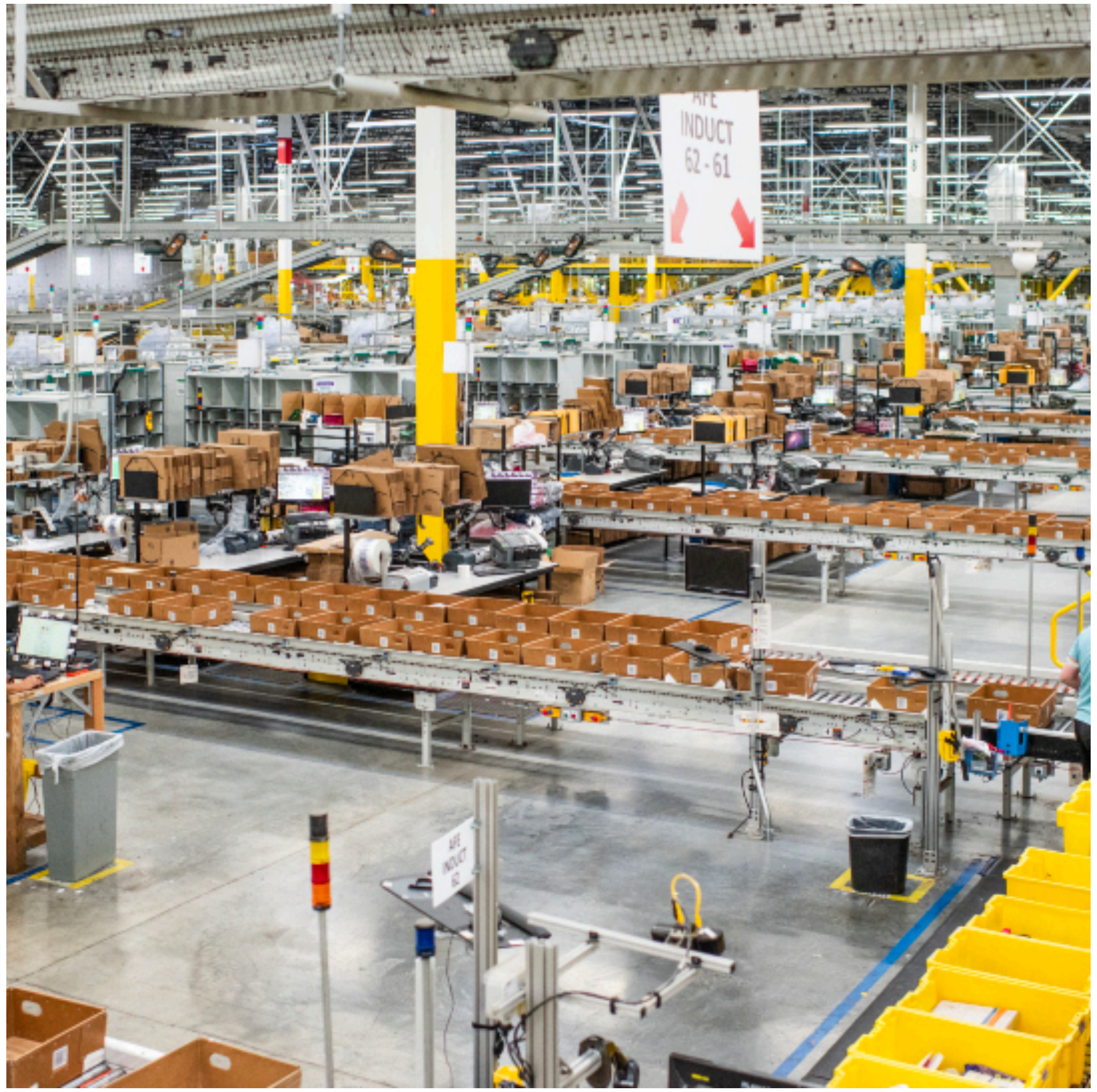
The Supply Chain Problem

The Supply Chain Problem

- Supply Chain is about buying, storing, and transporting goods.

The Supply Chain Problem

- Supply Chain is about buying, storing, and transporting goods.
- There is a lot of historical “off-policy” data
  - e.g. Amazon, …



The Supply Chain Problem

- Supply Chain is about buying, storing, and transporting goods.
- There is a lot of historical “off-policy” data
    - e.g. Amazon, …
- Today: how can we use this data to solve the inventory management problem?
    - counterfactual issues?

Outline

Can we use historical data to solve inventory management problems in supply chain?

- How to use historical data?
- Moving to real-world inventory management problems
- Real world results

Largely based on this paper:
arxiv/2210.03137

Deep Inventory Management

Dhruv Madeka
Amazon, maded@amazon.com

Kari Torkkola
Amazon, karito@amazon.com

Carson Eisenach
Amazon, ceisen@amazon.com

Anna Luo
Pinterest*, annaluo676@gmail.com

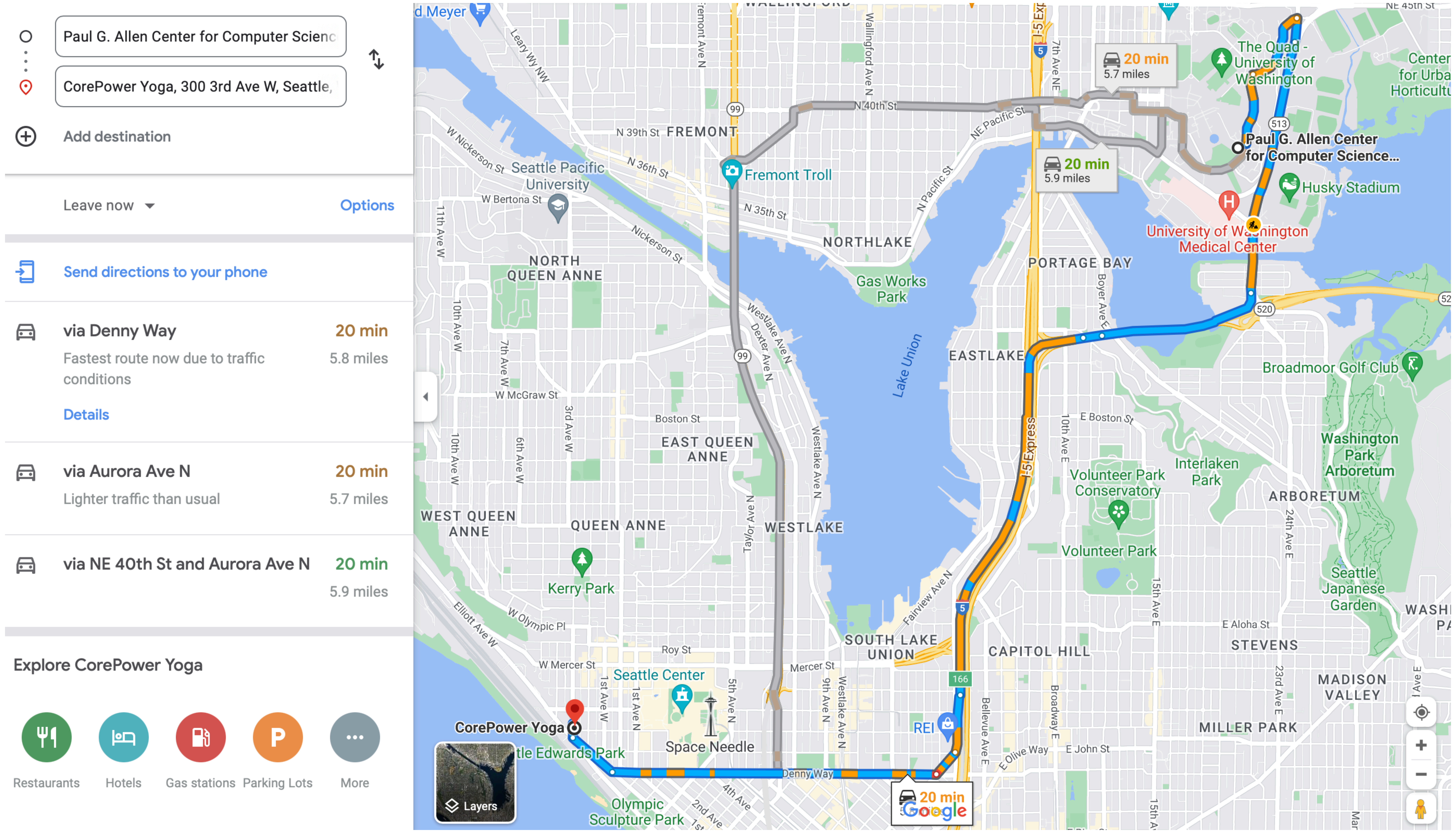
Dean P. Foster
Amazon, foster@amazon.com

Sham M. Kakade
Amazon, Harvard University, shamisme@amazon.com

I: Utilizing historical data

Warm up: Vehicle Routing

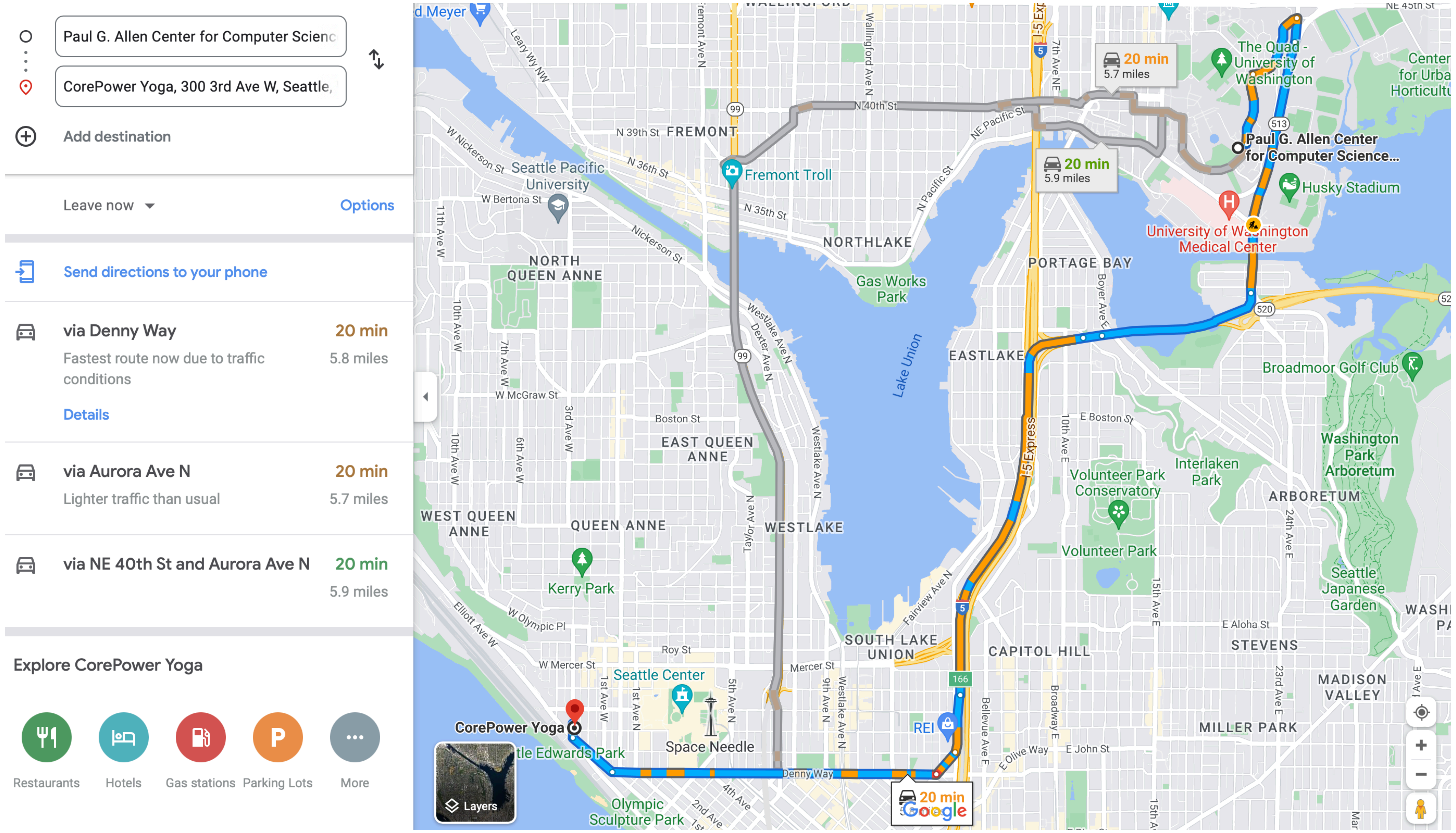
(when using historical data might be ok)



Warm up: Vehicle Routing

(when using historical data might be ok)

- We want a good policy for routing a single car.

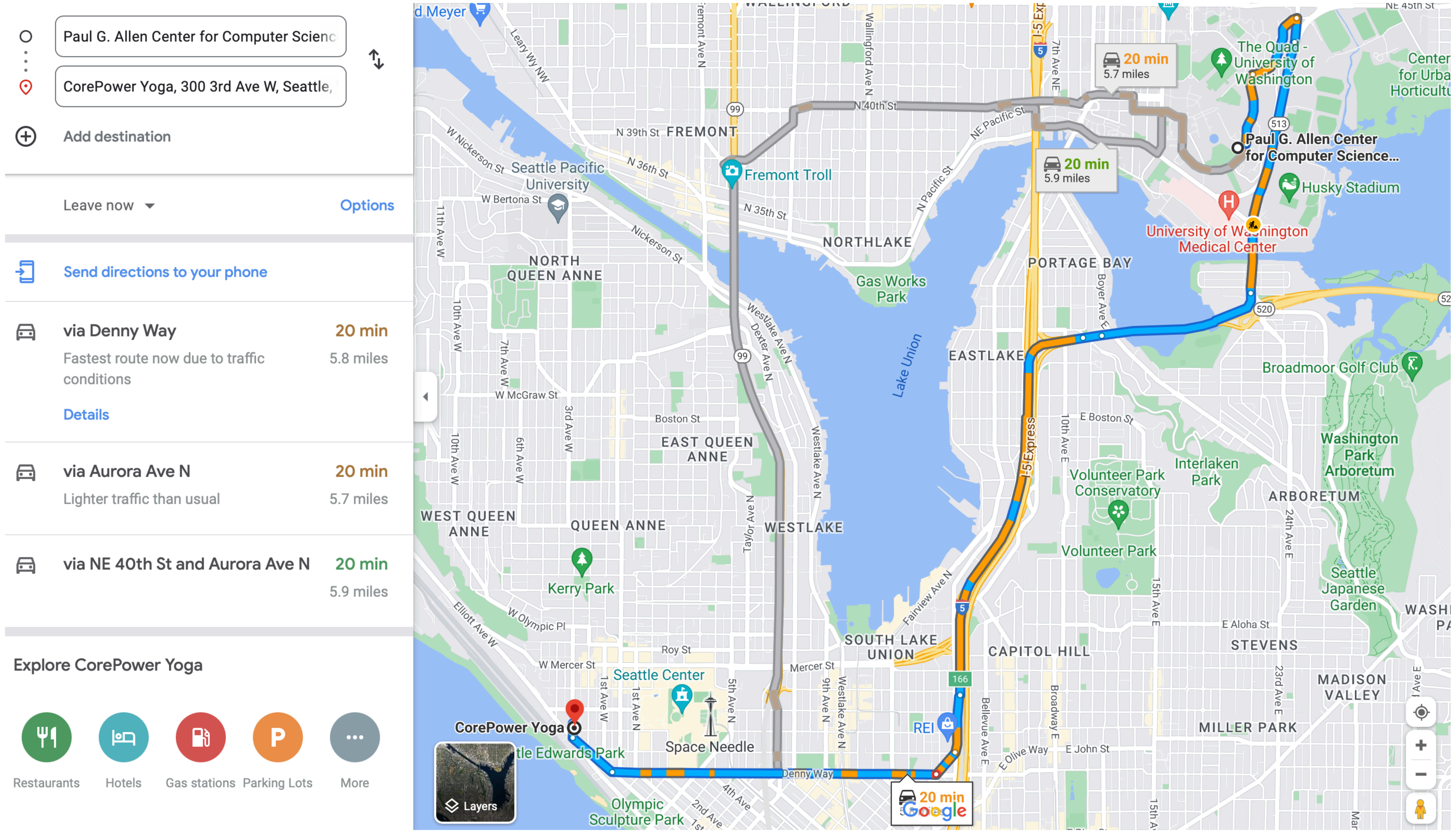


Warm up: Vehicle Routing

(when using historical data might be ok)

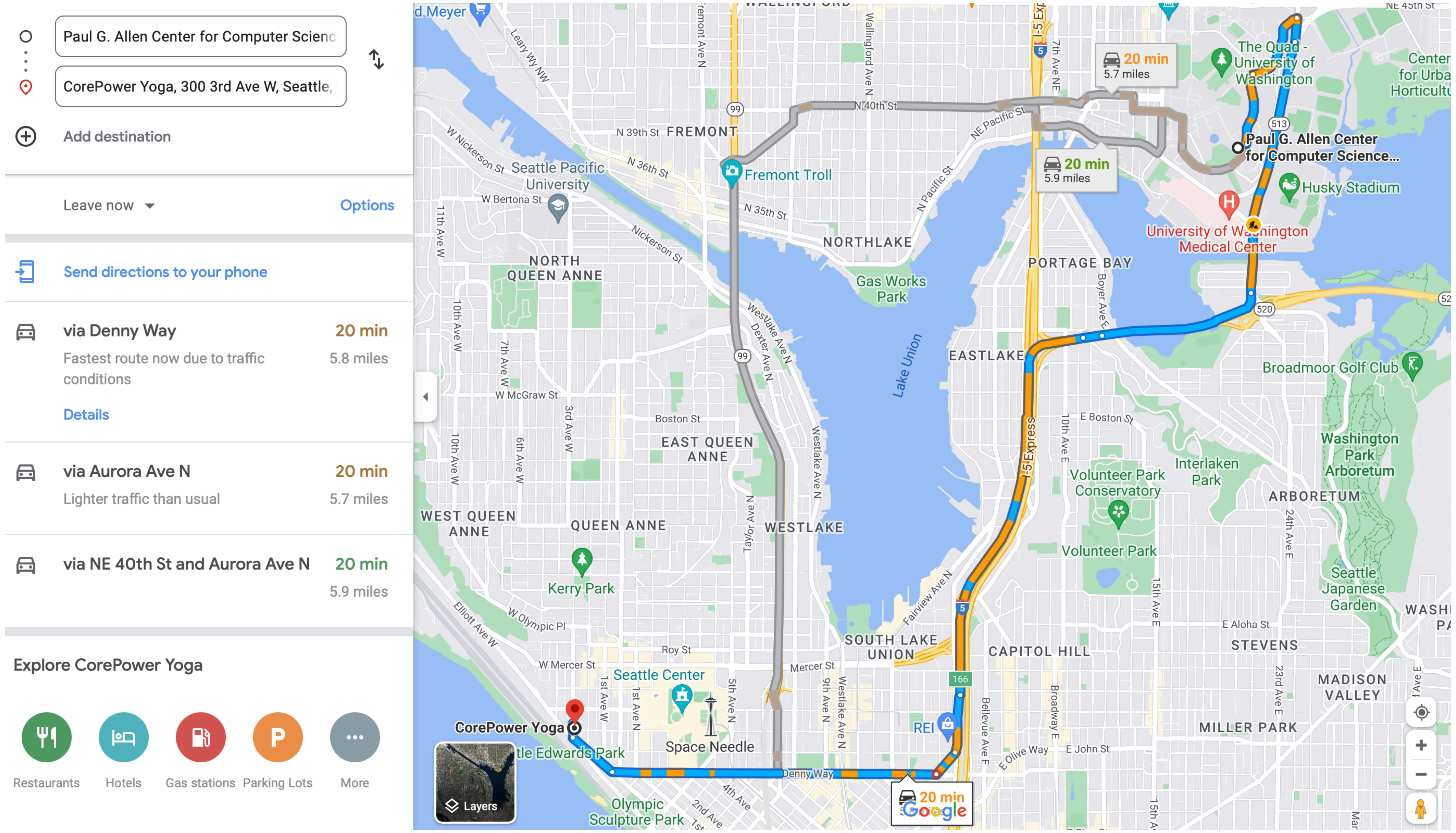
- We want a good policy for routing a single car.

- Policy π: features -> directions
  features: time of day, holiday indicators, current traffic, sports games, accidents, location, weather,



Warm up: Vehicle Routing

(when using historical data might be ok)



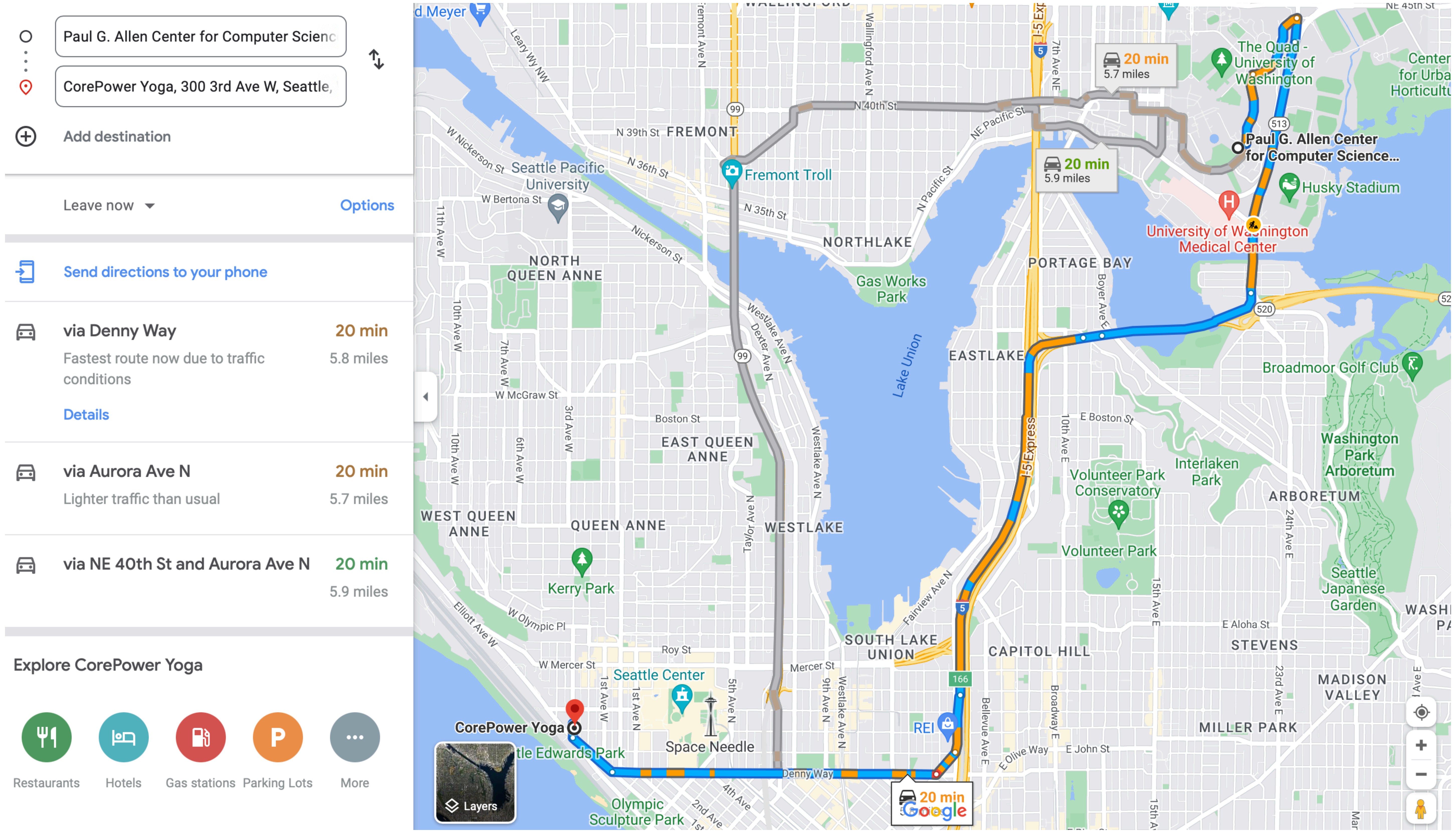
- We want a good policy for routing a single car.

- Policy π: features -> directions
  features: time of day, holiday indicators, current traffic, sports games, accidents, location, weather,

- Historical Data:
  suppose we have logged historical data of features

Warm up: Vehicle Routing

(when using historical data might be ok)



- We want a good policy for routing a single car.

- Policy π: features -> directions
  features: time of day, holiday indicators, current traffic, sports games, accidents, location, weather,

- Historical Data:
  suppose we have logged historical data of features

- Backtesting policies:
  - Key idea: a single route minimally affects traffic
  - Counterfactual: with the historical data, we can see what would have happened with another policy.

Warm up 2: Fleet Routing

Warm up 2: Fleet Routing

- We want to route a whole fleet of self-driving taxis.

Warm up 2: Fleet Routing

- We want to route a whole fleet of self-driving taxis.

- Policy π: features -> directions
  - features: customer demand, time, holiday indicators, current traffic, sports games, accidents, location, weather…

Warm up 2: Fleet Routing

- We want to route a whole fleet of self-driving taxis.

- Policy π: features -> directions
  - features: customer demand, time, holiday indicators, current traffic, sports games, accidents, location, weather…

- Historical Data:
  suppose we have logged historical data of features

Warm up 2: Fleet Routing

- We want to route a whole fleet of self-driving taxis.

- Policy π: features -> directions
  - features: customer demand, time, holiday indicators, current traffic, sports games, accidents, location, weather…

- Historical Data:
  suppose we have logged historical data of features

- Backtesting policies:
  - Key idea: a small fleet route may have small affects on traffic.
  - Counterfactual: with the historical data, we can see what would have happened with another policy.

Supply Chain Data

Supply Chain Data

Price= $2
Cost= $1

Time	Inventory	Demand	Order	Revenue

Supply Chain Data

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40

Price= $2
Cost= $1

Supply Chain Data

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	10	-10

Price= $2
Cost= $1

Supply Chain Data

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	10	-10
1	90	20	-	40

80+10

Price= $2

Cost= $1

Supply Chain Data

Price= $2
Cost= $1

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	10	-10
1	90	20	-	40
1	70	-	50	-50

Supply Chain Data

Price= $2
Cost= $1

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	10	-10
1	90	20	-	40
1	70	-	50	-50
2	120	60	-	120

Supply Chain Data

Price= $2
Cost= $1

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	10	-10
1	90	20	-	40
1	70	-	50	-50
2	120	60	-	120
2	60	-	10	-10

Backtesting a policy

Backtesting a policy

- Current order doesn't impact future demand.
  - This allows us to backtest!

Backtesting a policy

Time	Inventory	Demand	Order	Revenue

Price= $2

Cost= $1

- Current order doesn't impact future demand.
  - This allows us to backtest!

Backtesting a policy

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40

Price= $2
Cost= $1

- Current order doesn't impact future demand.
  - This allows us to backtest!

Backtesting a policy

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	~~10~~ 40	~~-10~~ -40

Price= $2
Cost= $1

- Current order doesn't impact future demand.
  - This allows us to backtest!

Backtesting a policy

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	~~10~~ 40	~~-10~~ -40
1	~~90~~ 120	20	-	40

Price= $2

Cost= $1

- Current order doesn't impact future demand.
  - This allows us to backtest!

Backtesting a policy

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	~~10~~ 40	~~-10~~ -40
1	~~90~~ 120	20	-	40
1	~~70~~ 100	-	~~50~~ 20	~~-50~~ -20

Price= $2

Cost= $1

- Current order doesn't impact future demand.
  - This allows us to backtest!

Backtesting a policy

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	~~10~~ 40	~~-10~~ -40
1	~~90~~ 120	20	-	40
1	~~70~~ 100	-	~~50~~ 20	~~-50~~ -20
2	120	60	-	120
2	60	-	10	-10

80 + 40

100 − 20

Price= $2

Cost= $1

- Current order doesn't impact future demand.
  - This allows us to backtest!

Backtesting a policy

Time	Inventory	Demand	Order	Revenue
0	100	20	-	40
0	80	-	~~10~~ 40	~~-10~~ -40
1	~~90~~ 120	20	-	40
1	~~70~~ 100	-	~~50~~ 20	~~-50~~ -20
2	120	60	-	120
2	60	-	10	-10

Price= $2
Cost= $1

- Current order doesn't impact future demand.
  - This allows us to backtest!
  - Empirically, backlog due to unmet demand does not look significant.[1]

1. See Verhoef et al (2006)

Formalization of the Supply Chain Problem

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]
- The supply chain problem as an ExoMDP:

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]
- The supply chain problem as an ExoMDP:
  - Action a_t: how much you buy

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]
- The supply chain problem as an ExoMDP:
  - Action a_t: how much you buy
  - Exogenous random variables: evolving under $\Pr$ and not dependent on our actions
    $(\text{Demand}_t, \text{Price}_t, \text{Cost}_t, \text{Lead Time}_t, \text{Covariates}_t) := s_t$

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]
- The supply chain problem as an ExoMDP:
  - Action a_t: how much you buy
  - Exogenous random variables: evolving under $\Pr$ and not dependent on our actions
    (Demand$_t$, Price$_t$, Cost$_t$, Lead Time$_t$, Covariates$_t$) $:= s_t$
  - Known controllable part (inventory) I_t: (known) evolution is dependent on our action.
    - $I_t = \max(I_{t-1} + a_{t-1} - D_t, 0)$ (and suppose we start at I_0).
  - Immediate reward is the profits: $r(s_t, I_t, a_t) :=$ Price$_t \times \min($Demand$_t, I_t) -$ Cost$_t \times a_t$

"rl state" : (I_t, s_t)

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]
- The supply chain problem as an ExoMDP:
  - Action a_t: how much you buy
  - Exogenous random variables: evolving under $\Pr$ and not dependent on our actions $(\text{Demand}_t, \text{Price}_t, \text{Cost}_t, \text{Lead Time}_t, \text{Covariates}_t) := s_t$
  - Known controllable part (inventory) I_t: (known) evolution is dependent on our action.
    - $I_t = \max(I_{t-1} + a_{t-1} - D_t, 0)$ (and suppose we start at I_0).
  - Immediate reward is the profits: $r(s_t, I_t, a_t) := \text{Price}_t \times \min(\text{Demand}_t, I_t) - \text{Cost}_t \times a_t$
- Learning setting:

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]
- The supply chain problem as an ExoMDP:
  - Action a_t: how much you buy
  - Exogenous random variables: evolving under $\Pr$ and not dependent on our actions (Demand$_t$, Price$_t$, Cost$_t$, Lead Time$_t$, Covariates$_t$) $:= s_t$
  - Known controllable part (inventory) I_t: (known) evolution is dependent on our action.
    - $I_t = \max(I_{t-1} + a_{t-1} - D_t, 0)$ (and suppose we start at I_0).
  - Immediate reward is the profits: $r(s_t, I_t, a_t) :=$ Price$_t \times \min($Demand$_t, I_t) -$ Cost$_t \times a_t$
- Learning setting:
  - Offline Data: We observe N historical trajectories, where each sequence is sampled $s_1, \ldots, s_H \sim \Pr$

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]
- The supply chain problem as an ExoMDP:
  - Action a_t: how much you buy
  - Exogenous random variables: evolving under $\Pr$ and not dependent on our actions $(\text{Demand}_t, \text{Price}_t, \text{Cost}_t, \text{Lead Time}_t, \text{Covariates}_t) := s_t$
  - Known controllable part (inventory) I_t: (known) evolution is dependent on our action.
    - $I_t = \max(I_{t-1} + a_{t-1} - D_t, 0)$ (and suppose we start at I_0).
  - Immediate reward is the profits: $r(s_t, I_t, a_t) := \text{Price}_t \times \min(\text{Demand}_t, I_t) - \text{Cost}_t \times a_t$
- Learning setting:
  - Offline Data: We observe N historical trajectories, where each sequence is sampled $s_1, \ldots, s_T \sim \Pr$
  - Goal: maximize our ~~over~~ H step cumulative reward:

Formalization of the Supply Chain Problem

- Exogenous MDPs: Growing literature around a class of MDPs where a large part of the state is driven by an exogenous noise process [Efroni et al 2021, Sinclair et al 2022]
- The supply chain problem as an ExoMDP:
  - Action a_t: how much you buy
  - Exogenous random variables: evolving under $\Pr$ and not dependent on our actions $(\text{Demand}_t, \text{Price}_t, \text{Cost}_t, \text{Lead Time}_t, \text{Covariates}_t) := s_t$
  - Known controllable part (inventory) I_t: (known) evolution is dependent on our action.
    - $I_t = \max(I_{t-1} + a_{t-1} - D_t, 0)$ (and suppose we start at I_0).
  - Immediate reward is the profits: $r(s_t, I_t, a_t) := \text{Price}_t \times \min(\text{Demand}_t, I_t) - \text{Cost}_t \times a_t$
- Learning setting:
  - Offline Data: We observe N historical trajectories, where each sequence is sampled $s_1, \ldots, s_T \sim \Pr$
  - Goal: maximize our over H step cumulative reward:

$$V_H(\pi) = E_\pi\left[\sum_{t=1}^{H} \gamma^t r(s_t, I_t, a_t)\right]$$

Why is it an interesting RL problem?

Why is it an interesting RL problem?

- Lots of time dependence!
  - If you buy too much, you're left with the inventory for months!
  - Your actions (orders) affect the state at a random time later
  - Tons of correlation across time (Demand, Price, Cost, Seasonality, etc)

What do ExoMDPs buy us?

We can backtest (assuming the “controllable” dynamics are known) and avoid the counterfactual/causality issue!

What do ExoMDPs buy us?

We can backtest (assuming the "controllable" dynamics are known) and avoid the counterfactual/causality issue!

Theorem: RL in ExoMDPs is as easy as Supervised Learning

Suppose we have K policies $\Pi = \{\pi_1, \ldots \pi_K\}$, and we have N sampled exogenous paths. Then we can accurately backtest up to nearly $K \approx 2^N$ policies.

Formally, for $\delta \in (0,1)$, with pr. greater than $1 - \delta$ - we have that for all $\pi \in \Pi$:

$$|V_0(\pi) - \hat{V}_0(\pi)| \leq H\sqrt{\frac{\log(K/\delta)}{N}}$$

(assuming the reward r_t is bounded by 1).

What do ExoMDPs buy us?

We can backtest (assuming the "controllable" dynamics are known) and avoid the counterfactual/causality issue!

Theorem: RL in ExoMDPs is as easy as Supervised Learning

Suppose we have K policies $\Pi = \{\pi_1, \ldots \pi_K\}$, and we have N sampled exogenous paths. Then we can accurately backtest up to nearly $K \approx 2^N$ policies.

Formally, for $\delta \in (0,1)$, with pr. greater than $1 - \delta$ - we have that for all $\pi \in \Pi$:

$$| V_0(\pi) - \hat{V}_0(\pi) | \leq H\sqrt{\frac{\log(K/\delta)}{N}}$$

(assuming the reward r_t is bounded by 1).

- Implications:
  - We can optimize a neural policy on the past data.
  - In the usual RL setting (not exogenous), we would have an amplification factor of (at least) $\min\{2^H, K\}$, using historical data due to the counterfactual issue.

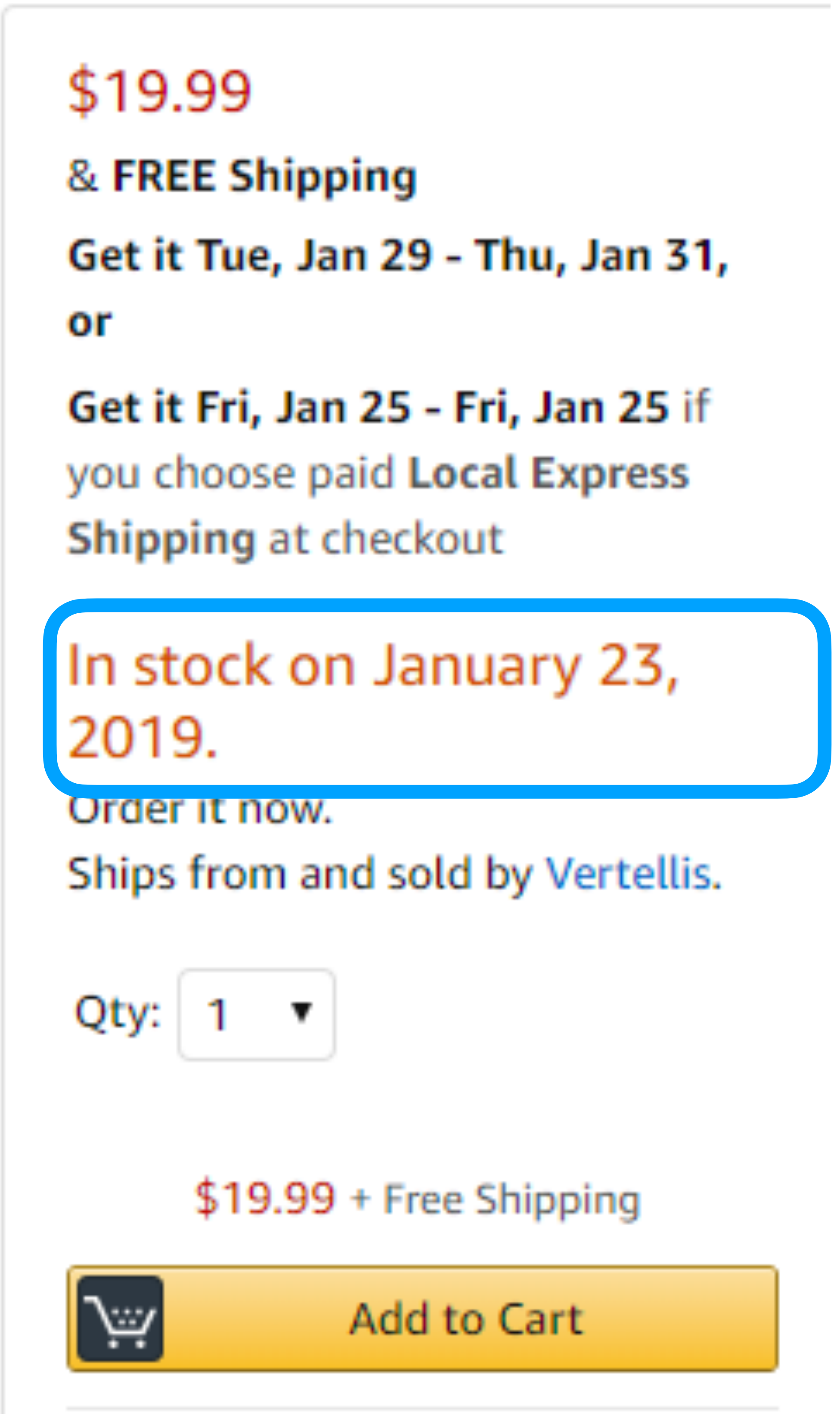
II: Real World Inventory Management Problems

Real-world Issue: Censored Demand

- When demand $\geq$ inventory, what customers see:

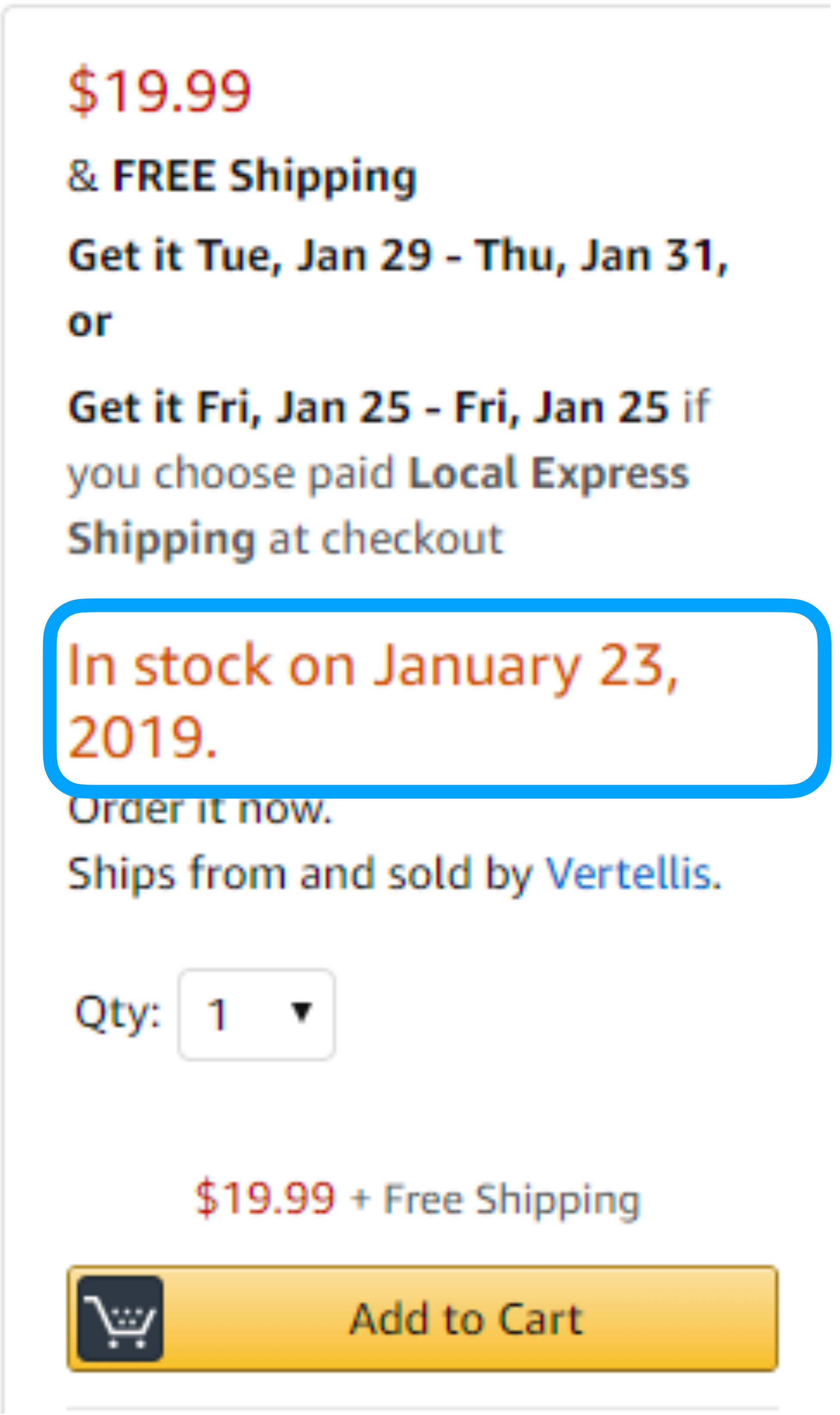
Real-world Issue: Censored Demand

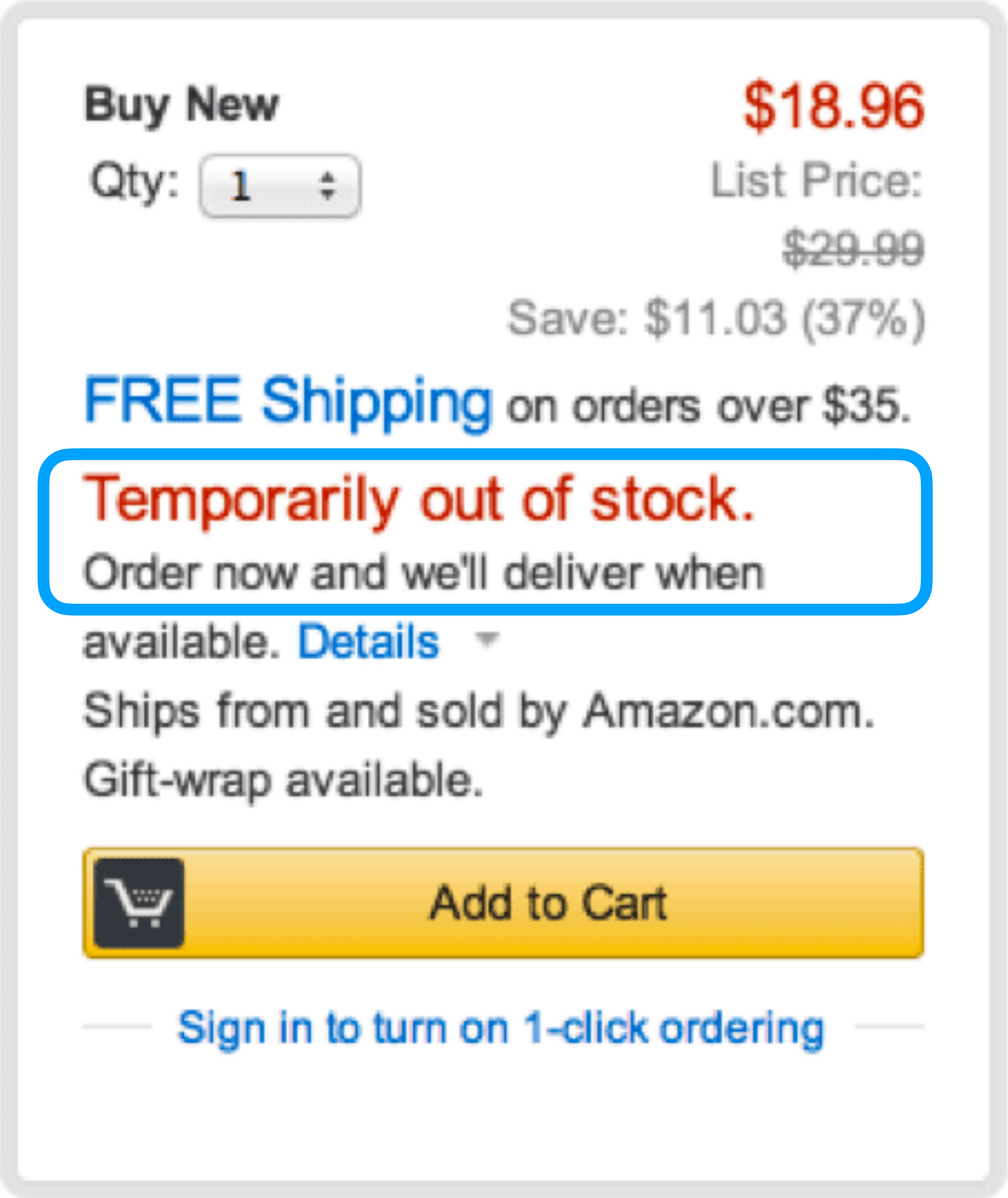
- When demand $\geq$ inventory, what customers see:



Real-world Issue: Censored Demand

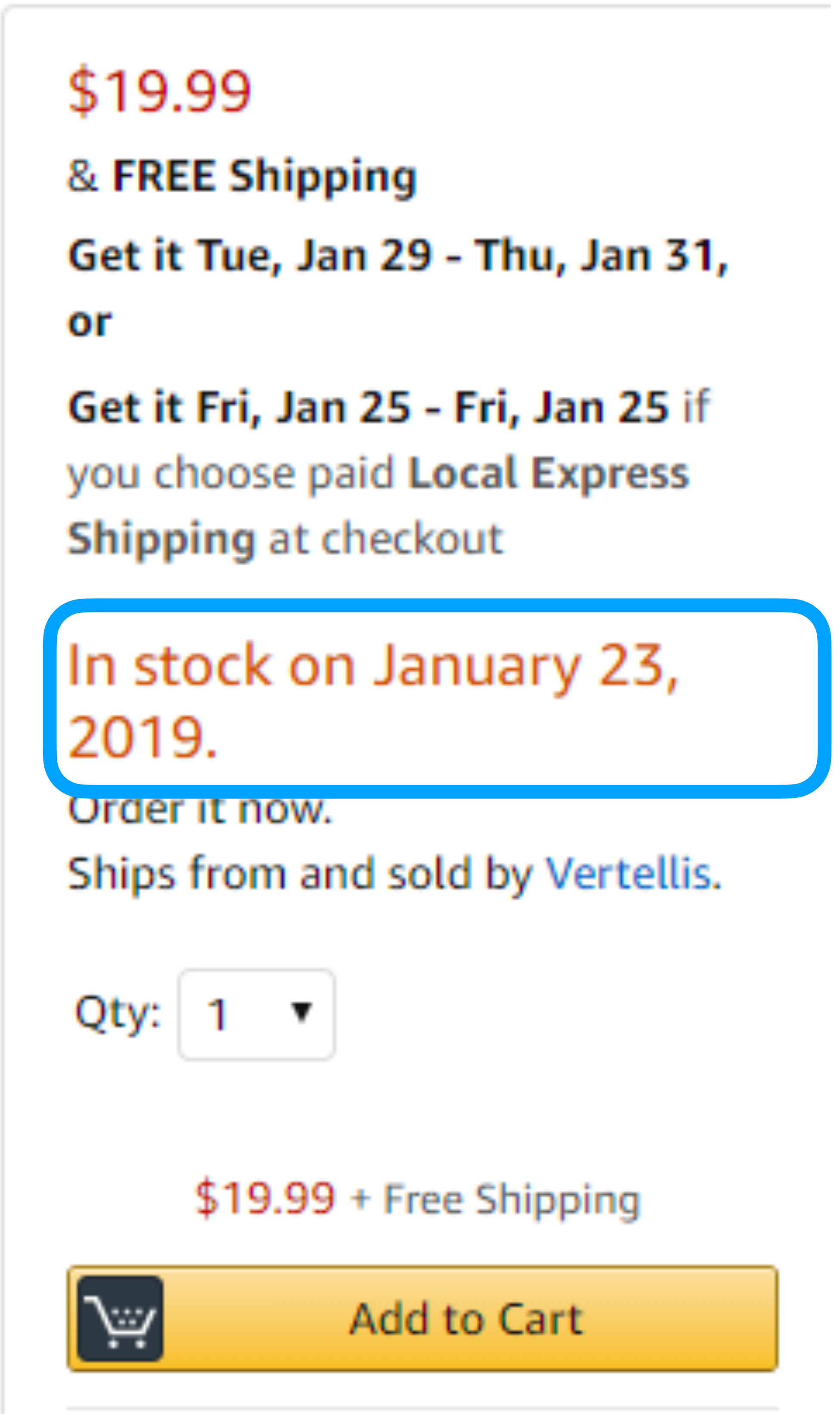
- When demand $\geq$ inventory, what customers see:

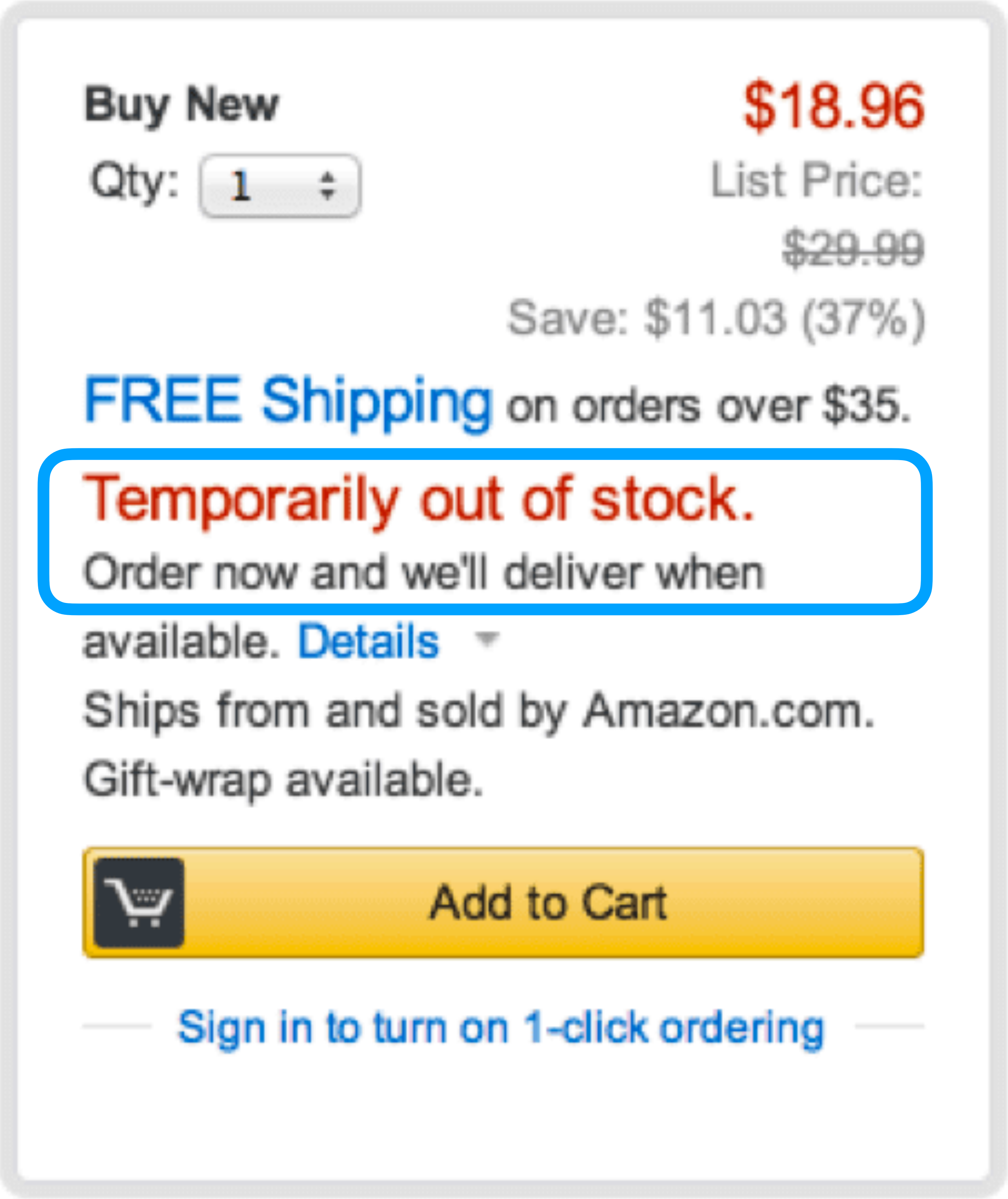




Real-world Issue: Censored Demand

- When demand $\geq$ inventory, what customers see:





We only observe **sales** not the **demand:**

Sales $:=$ min(Demand, Inventory)

Can we still backtest?

Our historical data is then censored….

$$\text{Sales} := \min(\text{Demand}, \text{Inventory})$$

Price= $2

Cost= $1

Our historical data is then censored….

$$\text{Sales} := \min(\text{Demand}, \text{Inventory})$$

Price= $2
Cost= $1

Time	Inventory	True Demand	Sales	Order	Revenue
T	10	??	10	-	20
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮

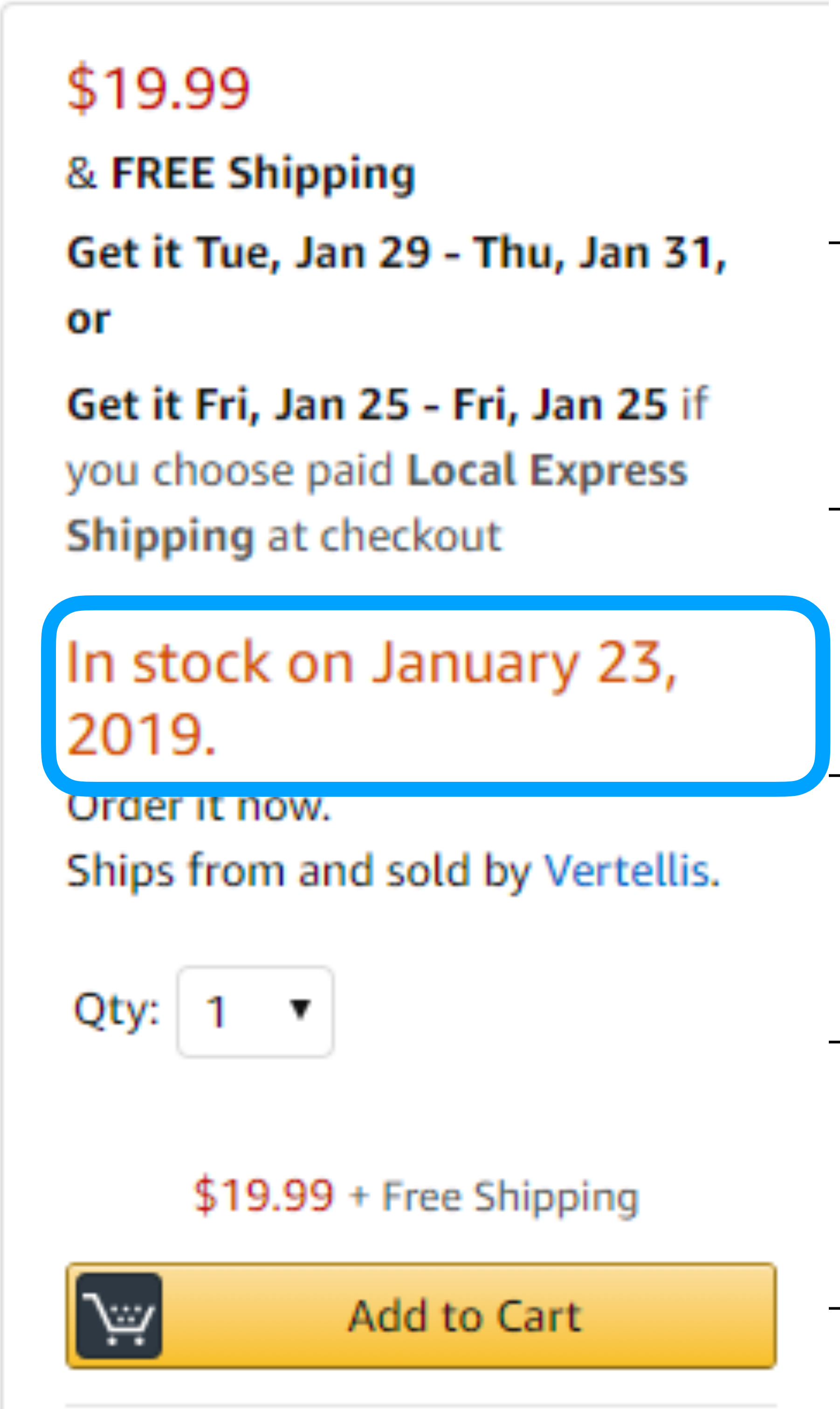
Our historical data is then censored….

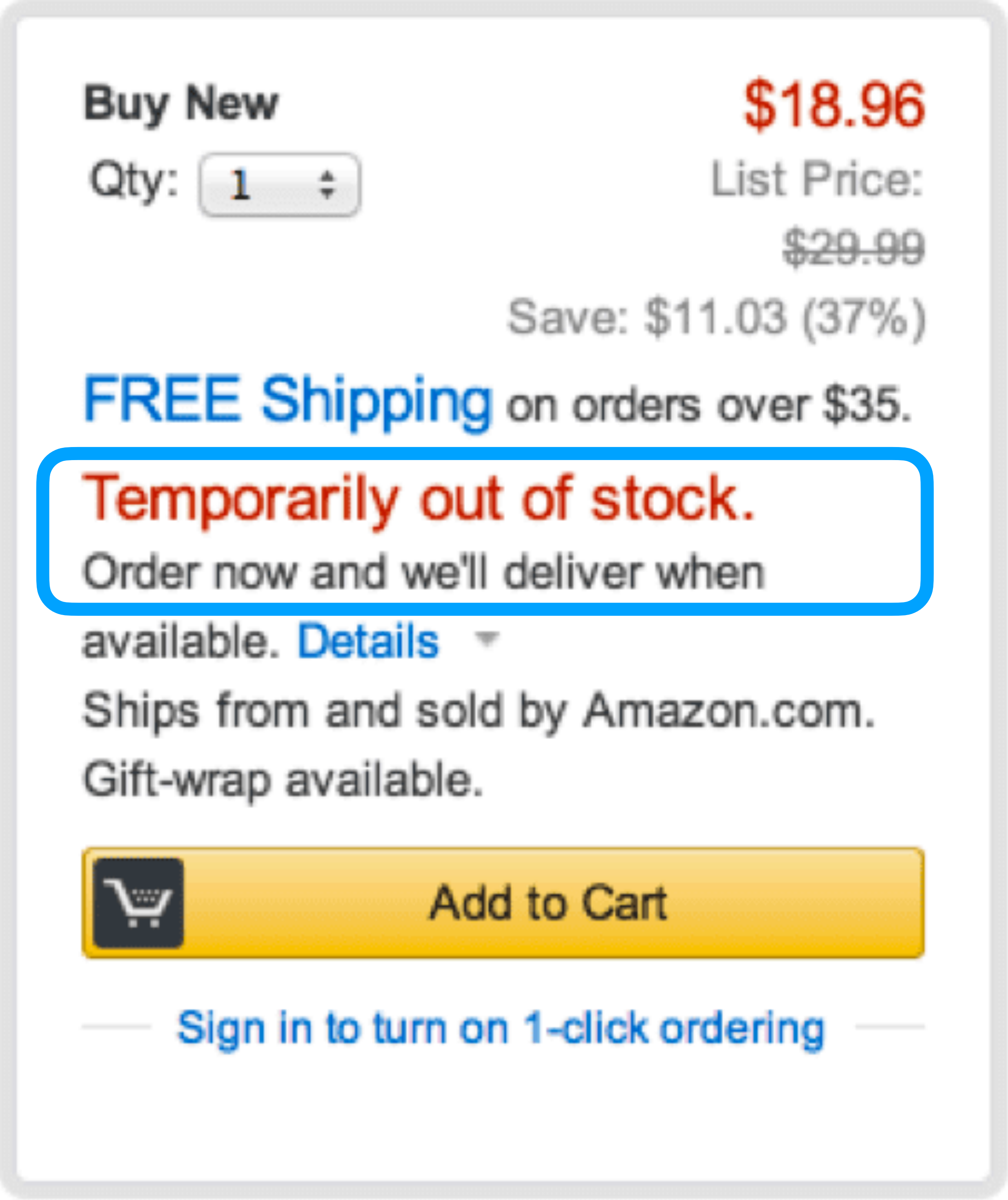
$$\text{Sales} := \min(\text{Demand}, \text{Inventory})$$

Price= $2

Cost= $1

Time	Inventory	True Demand	Sales	Order	Revenue
T	10	??	10	-	20
⋮	⋮				
⋮	⋮				
⋮	⋮				
⋮	⋮				
⋮	⋮				





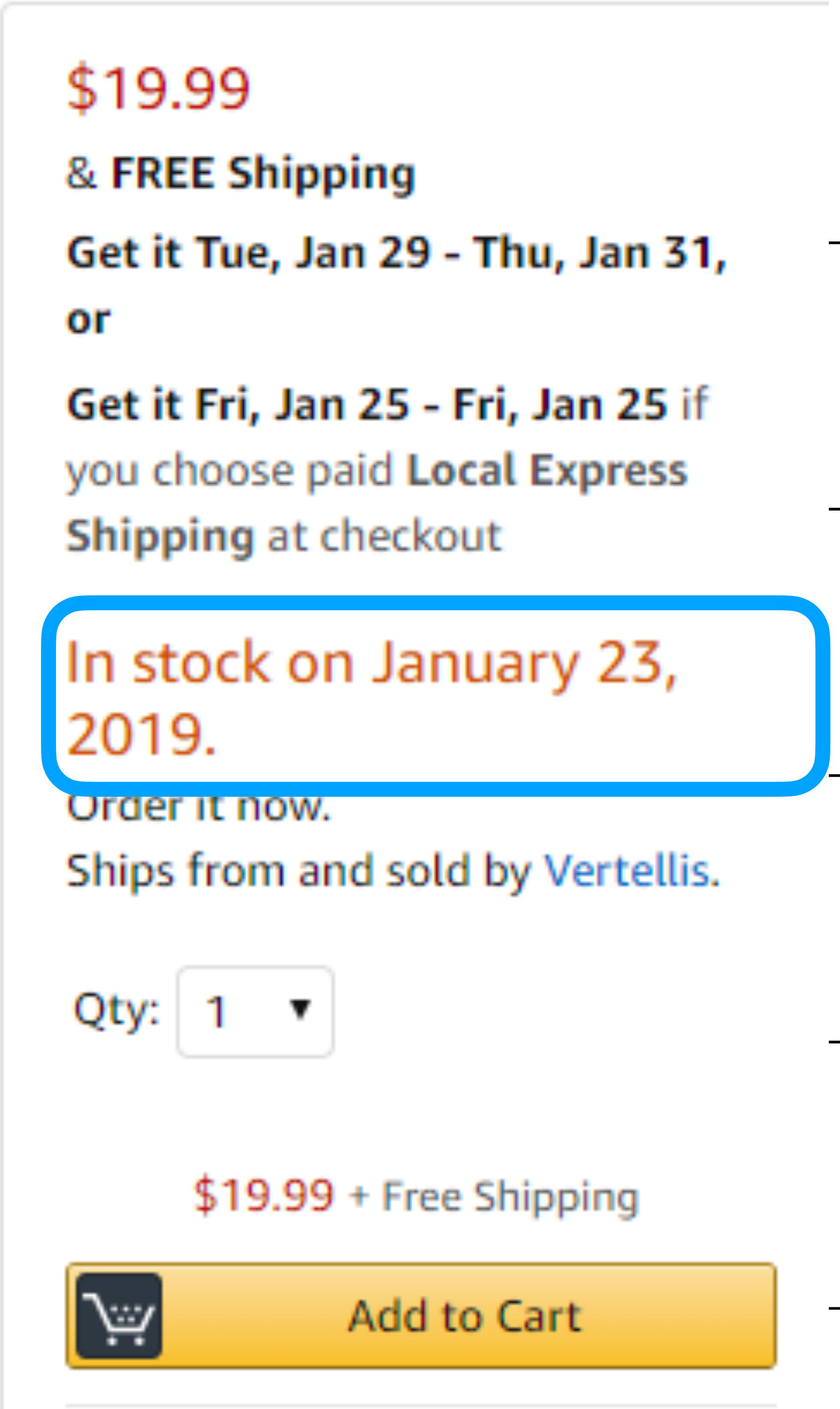
Our historical data is then censored….

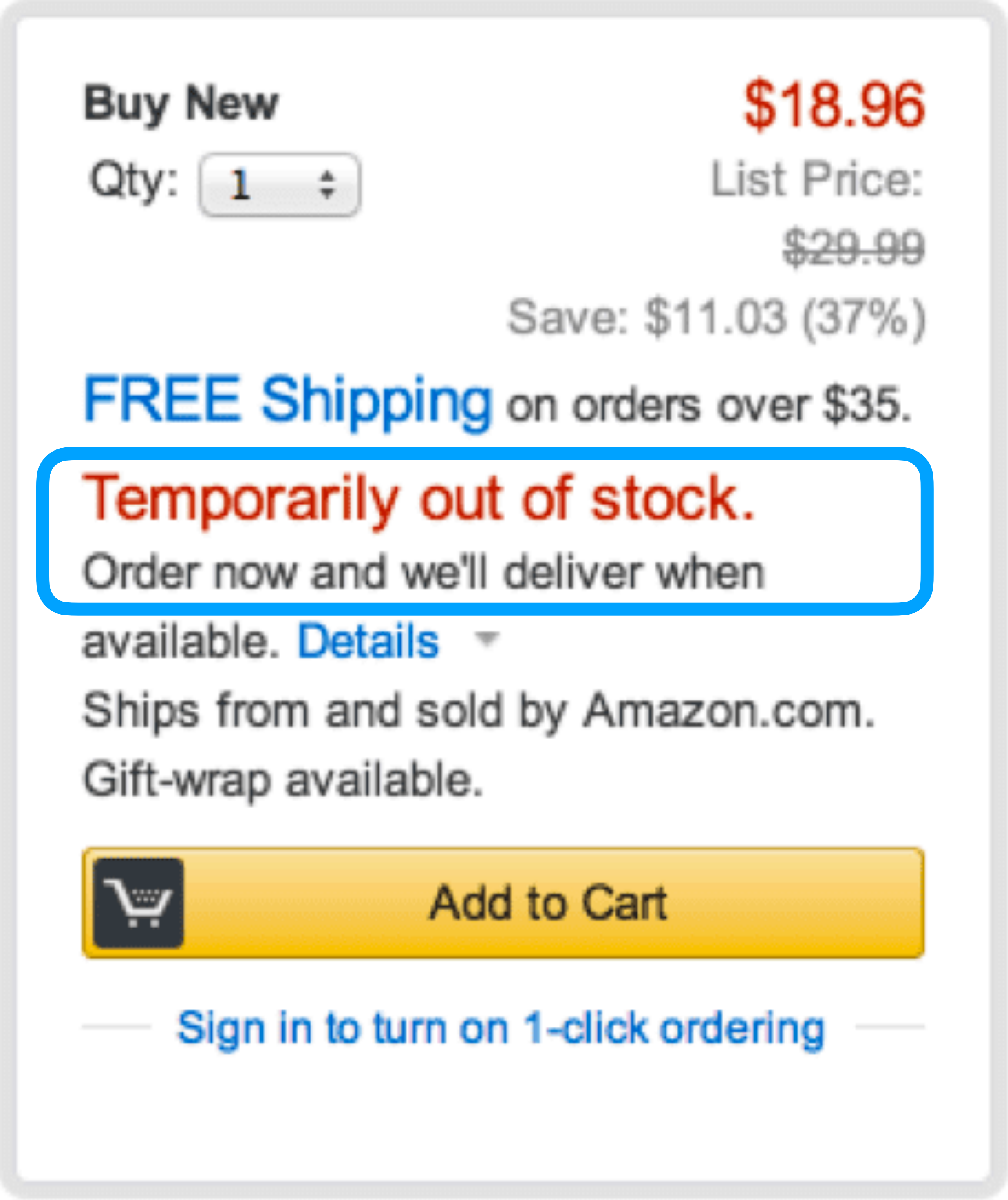
$$\text{Sales} := \min(\text{Demand}, \text{Inventory})$$

Time	Inventory	True Demand	Sales	Order	Revenue
T	10	??	10	-	20
⋮	⋮				
⋮	⋮				
⋮	⋮				
⋮	⋮				
⋮	⋮				

Price= $2
Cost= $1

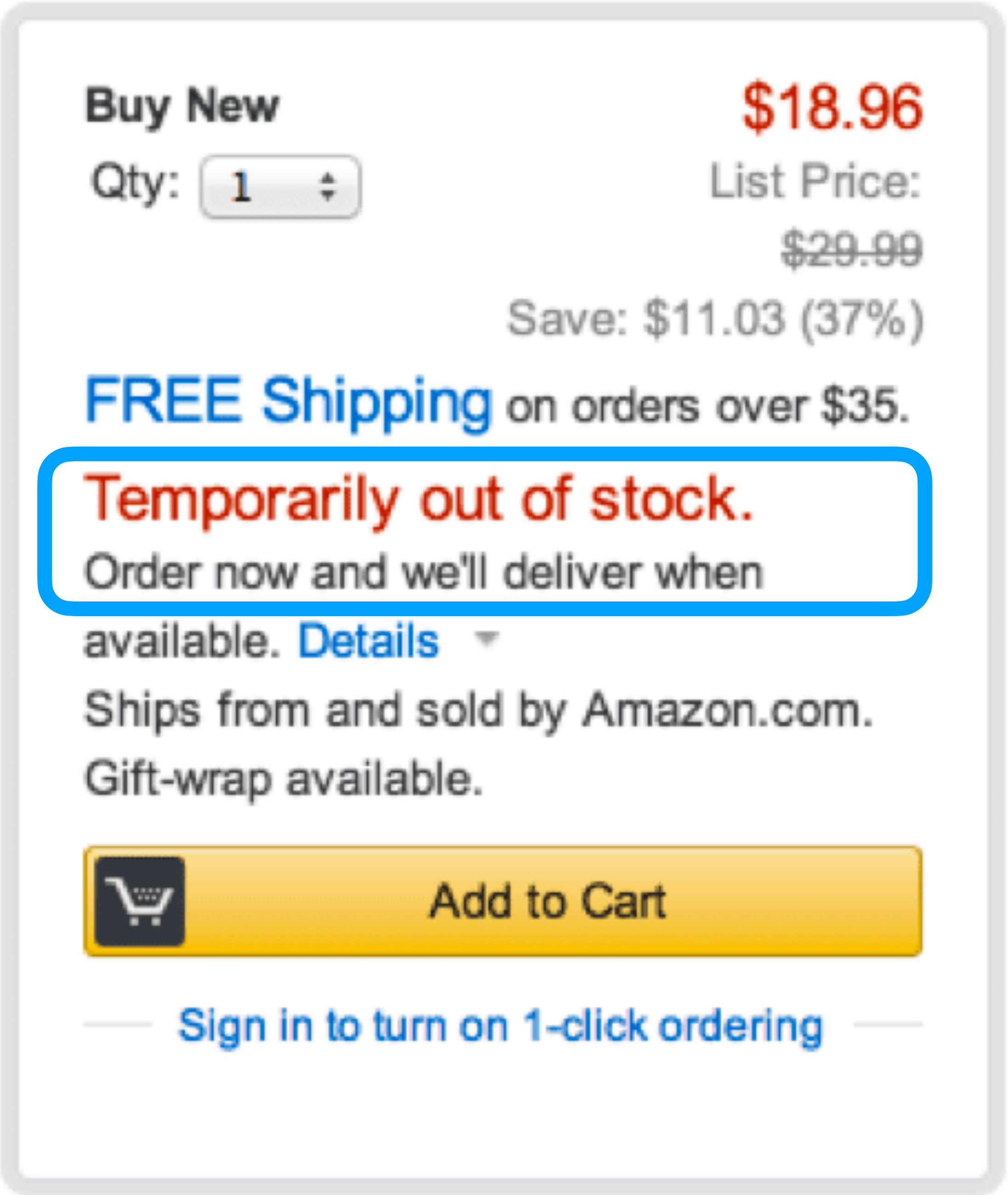
If we could fill in the missing demand, then we could still backtest!





We have many observed historical covariates

- Covariates:
  Sales, Web Site, Glance Views, Product Text, Reviews
- Example: the #times customers look at an item gives us info about the unobserved demand.

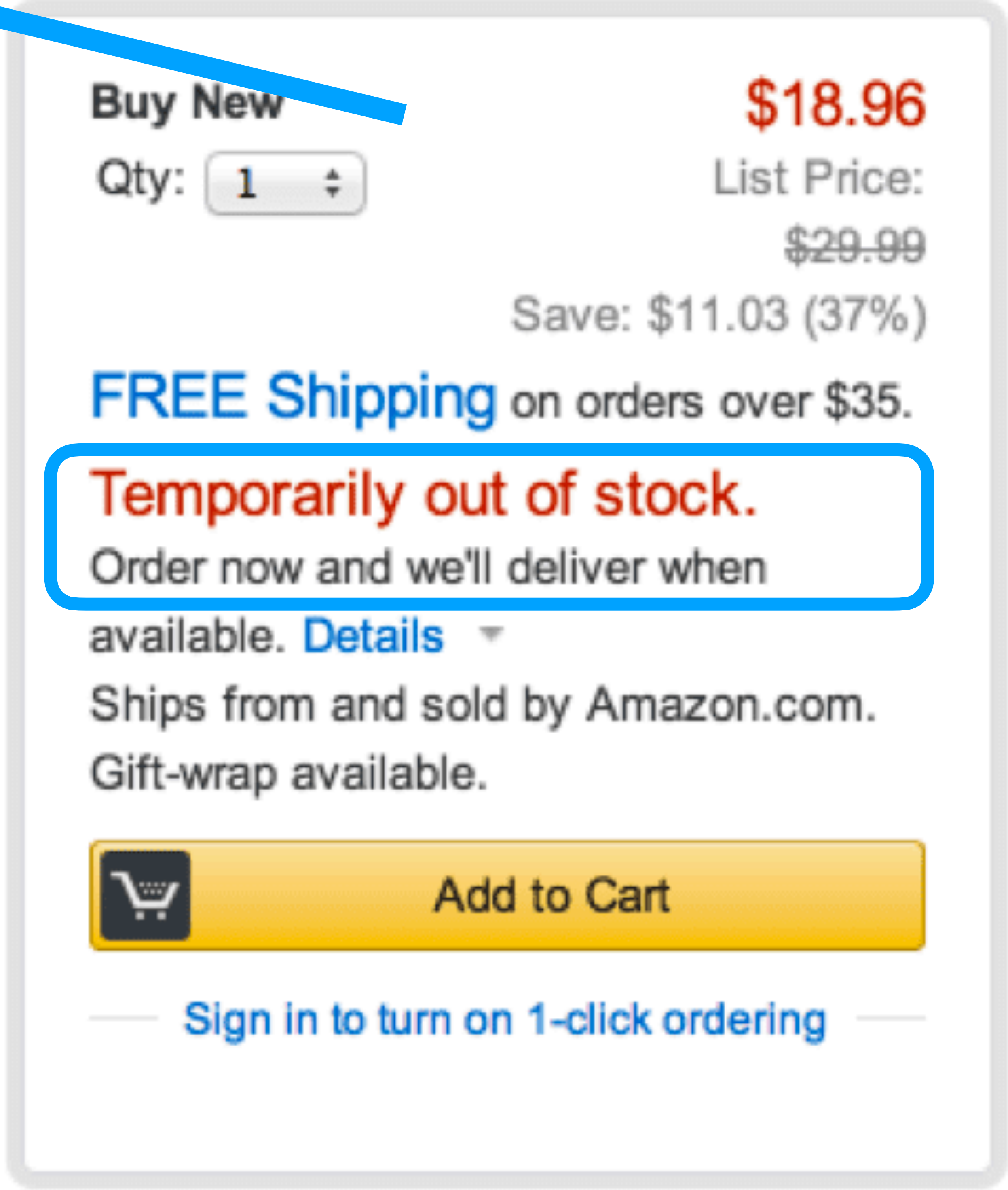


- Let's forecast the missing variables from the observed covariates!
  $\hat{\mathbb{P}}(\text{Missing Data} \mid \text{Observed Data})$

Uncensoring the data….

Sales := min(Demand, Inventory)

Price= $2
Cost= $1

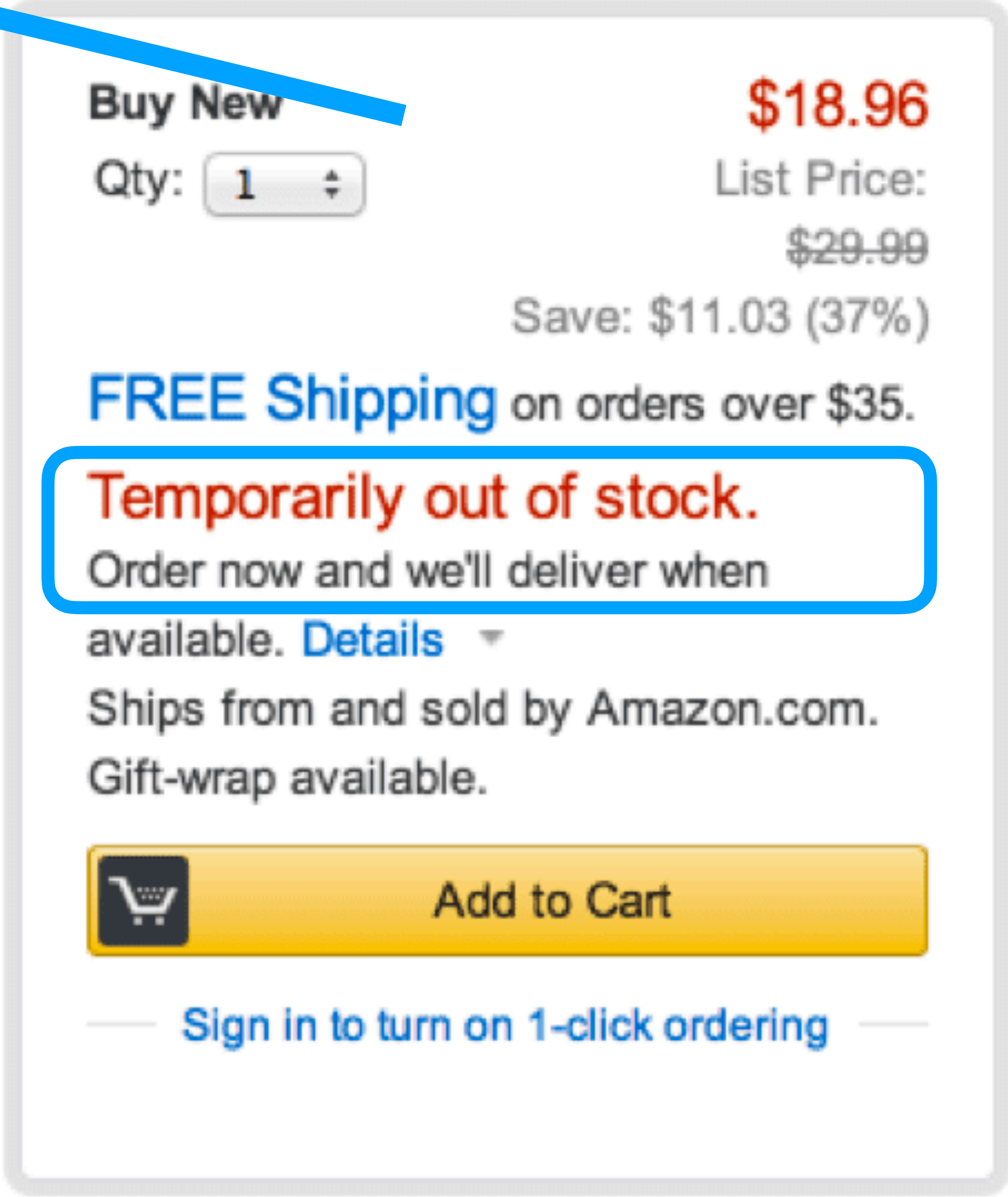


Uncensoring the data….

Sales := min(Demand, Inventory)

Price= $2
Cost= $1

Time	Inventory	True Demand	Sales	Order	Revenue
T	10	**40**	10	-	20
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		

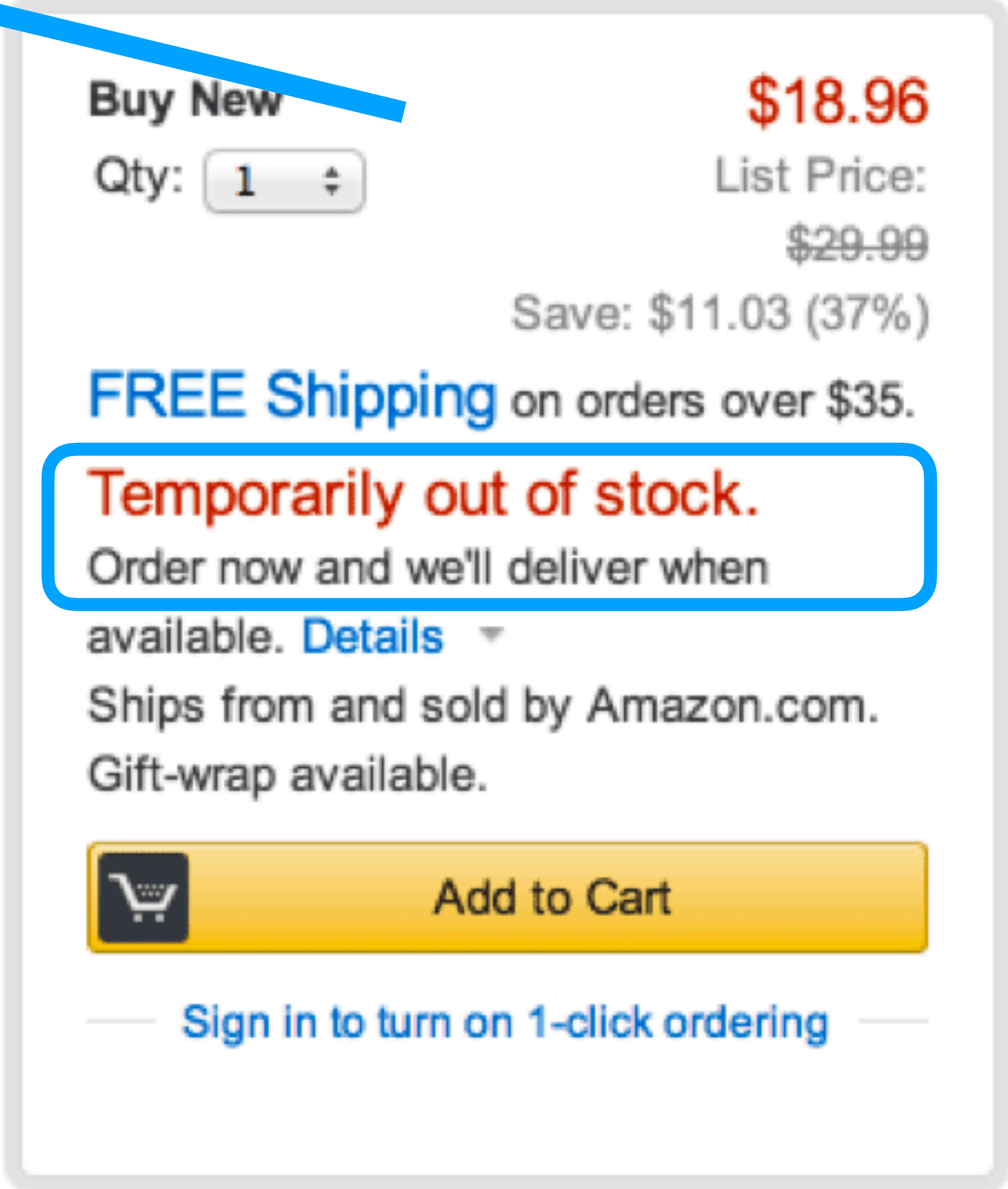


Uncensoring the data….

Sales := min(Demand, Inventory)

Price= $2
Cost= $1

Time	Inventory	True Demand	Sales	Order	Revenue
T	10	40	10	-	20
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		
⋮	⋮	⋮	⋮		



Key idea: Use covariates (e.g. glance views) to forecast missing demand, vendor lead times, etc

What do ExoMDPs buy us?

We can backtest (even with censored data) and avoid the counterfactual/causality issue!

What do ExoMDPs buy us?

We can backtest (even with censored data) and avoid the counterfactual/causality issue!

Theorem: If we can accurately forecast the missing (exo) variables (i.e. our SL error is small), then we can backtest accurately.

(with only additive error increase based on our SL error).

What do ExoMDPs buy us?

We can backtest (even with censored data) and avoid the counterfactual/causality issue!

Theorem: If we can accurately forecast the missing (exo) variables (i.e. our SL error is small), then we can backtest accurately.

(with only additive error increase based on our SL error).

Setting: we have N sampled sequences $\{s_1^i, s_2^i, \ldots s_H^i\}_{i=1}^N$,
where M_i and O_i are the missing and observed exogenous variables in sequence i.

Forecast: $\widehat{\mathbb{P}}^i = \widehat{\Pr}(M_i \mid O_i)$ is our forecast of $\mathbb{P}^i = \Pr(M_i \mid O_i)$.

Assume: With pr. 1, forecasting has low error: $\frac{1}{N}\sum_{i=1}^{N} \text{TotalVar}\left(\mathbb{P}^i, \widehat{\mathbb{P}}^i\right) \le \epsilon_{\text{sup}}$.

Guarantee: For any $\delta \in (0,1)$, with pr. greater than $1-\delta$, for all $\pi \in \Pi$:

$$|V_0(\pi) - \hat{V}_0(\pi)| \le H\left(\epsilon_{\text{sup}} + \sqrt{\frac{\log(K/\delta)}{N}}\right)$$

III: Training Policies & Empirical Results

The Simulator

The Simulator

- Collection of historical trajectories:
  - 1 million products
  - 104 weeks of data per product

Data

The Simulator

- Collection of historical trajectories:
  - 1 million products
  - 104 weeks of data per product
- Uncensoring:
  - Demand
  - Vendor Lead Times

← Data

← Corrections

The Simulator

- Collection of historical trajectories:
  - 1 million products
  - 104 weeks of data per product

- Uncensoring:
  - Demand
  - Vendor Lead Times

- Policy gradient methods in a "gym":
  - "gym" ↔ backtesting ↔ simulator
    (note the "simulator" isn't a good world model).
  - The policy can depend on many features.
    (seasonality, holiday indicators, demand history, product details, text features)

Data

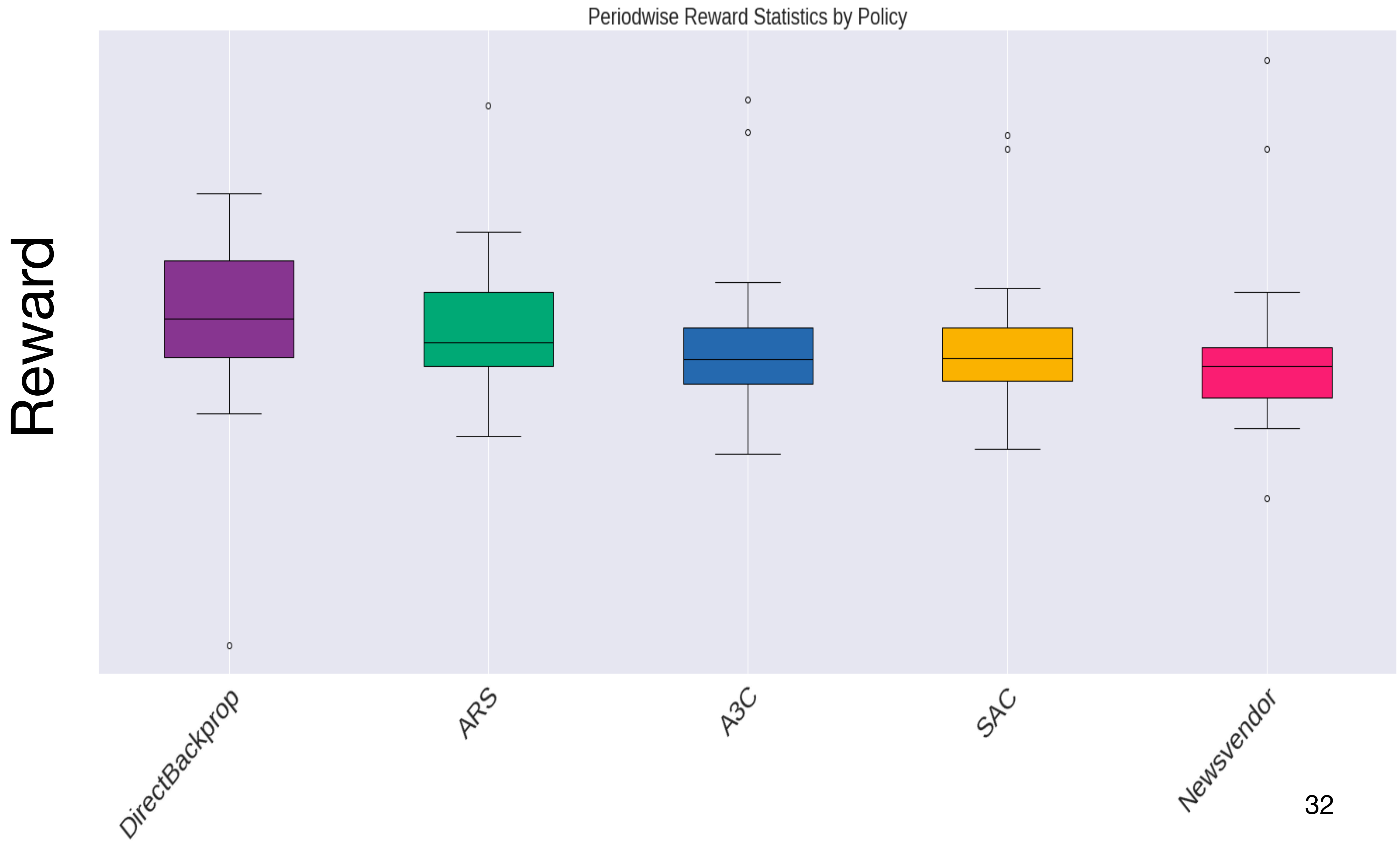
Corrections

Simulator

Sim to Real Transfer

- Sim: the backtest of DirectBackprop improves on Newsvendor.
- Real: DirectBackprop significantly reduces inventory without significantly reducing total revenue.

Simulation

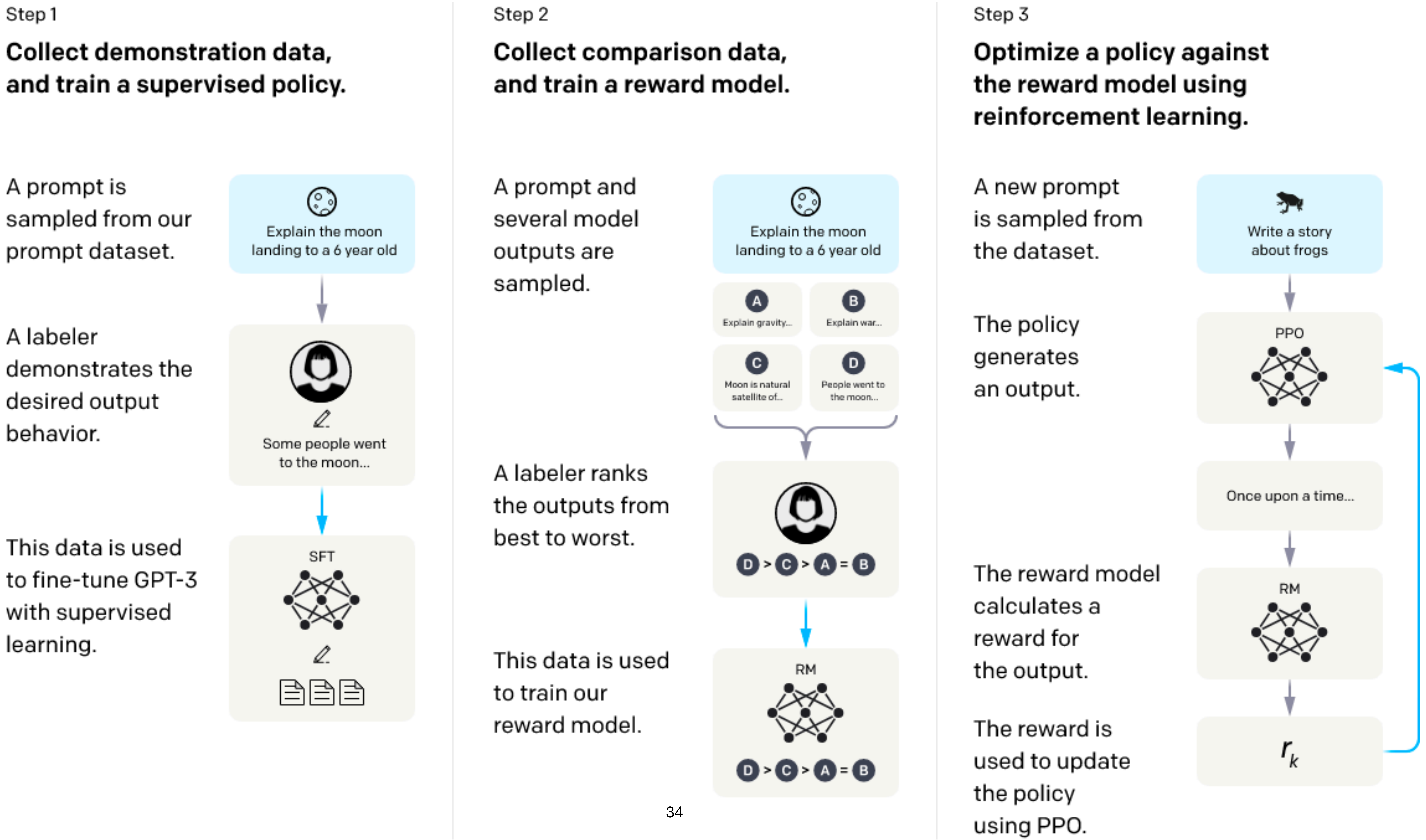


Real World

Metrics	% change
Inventory Level	**-12** $\pm$ 6
Revenue	**2.6%**

RLHF

RL from Human Feedback (RLHF)



Summary:

Attendance:

bit.ly/3RcTC9T

Feedback:

bit.ly/3RHtlxy

Summary:

Today: adding context to bandits requires SL but makes it much more useful

Attendance:

bit.ly/3RcTC9T

Feedback:

bit.ly/3RHtlxy

Summary:

Today: adding context to bandits requires SL but makes it much more useful

- The Course: sequential decision making (causality + decisions)
    - RL gives a helpful set of tools.
    - RL also gives an interesting viewpoint.

Attendance:
bit.ly/3RcTC9T

Feedback:
bit.ly/3RHtlxy

Summary:

Today: adding context to bandits requires SL but makes it much more useful

- The Course: sequential decision making (causality + decisions)
  - RL gives a helpful set of tools.
  - RL also gives an interesting viewpoint.
- **We hope you enjoyed the course!**

Attendance:
bit.ly/3RcTC9T

Feedback:
bit.ly/3RHtlxy

Extensions

Extensions

1. Can always replace contexts x_t with any fixed (vector-valued) function $\phi(x_t)$

Extensions

1. Can always replace contexts x_t with any fixed (vector-valued) function $\phi(x_t)$

   E.g., if believe rewards quadratic in scalar x_t, could make $\phi(x_t) = (x_t, x_t^2)$

Extensions

1. Can always replace contexts x_t with any fixed (vector-valued) function $\phi(x_t)$

   E.g., if believe rewards quadratic in scalar x_t, could make $\phi(x_t) = (x_t, x_t^2)$

2. Instead of fitting different $\theta^{(k)}$ for each arm, we could assume the mean reward is linear in some function of both the context and the action, i.e.,

$$\mathbb{E}_{r \sim \nu^{a_t}(x_t)}[r] = \phi(x_t, a_t)^\top \theta$$

Extensions

1. Can always replace contexts x_t with any fixed (vector-valued) function $\phi(x_t)$

   E.g., if believe rewards quadratic in scalar x_t, could make $\phi(x_t) = (x_t, x_t^2)$

2. Instead of fitting different $\theta^{(k)}$ for each arm, we could assume the mean reward is linear in some function of both the context and the action, i.e.,

$$\mathbb{E}_{r \sim \nu^{a_t}(x_t)}[r] = \phi(x_t, a_t)^\top \theta$$

   This is what problem 3 of HW 1 (which we cut) was about; it's helpful especially when K is large, since in that case there are a lot of $\theta^{(k)}$ to fit

Extensions

1. Can always replace contexts x_t with any fixed (vector-valued) function $\phi(x_t)$

   E.g., if believe rewards quadratic in scalar x_t, could make $\phi(x_t) = (x_t, x_t^2)$

2. Instead of fitting different $\theta^{(k)}$ for each arm, we could assume the mean reward is linear in some function of both the context and the action, i.e.,

$$\mathbb{E}_{r \sim \nu^{a_t}(x_t)}[r] = \phi(x_t, a_t)^\top \theta$$

   This is what problem 3 of HW 1 (which we cut) was about; it's helpful especially when K is large, since in that case there are a lot of $\theta^{(k)}$ to fit

Both cases allow a version of linUCB by extension of the same ideas: fit coefficients via least squares and use Chebyshev-like uncertainty quantification to get UCB

More detail on the combined linear model

For $t = 0 \rightarrow T - 1$

More detail on the combined linear model

For $t = 0 \to T - 1$

1. $\forall k$, define $A_t = \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda I$ and $\hat{\theta}_t = A_t^{-1} \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau) r_\tau$

More detail on the combined linear model

For $t = 0 \to T-1$

1. $\forall\, k$, define $A_t = \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda I$ and $\hat{\theta}_t = A_t^{-1} \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau) r_\tau$

2. Observe x_t & choose $a_t = \arg\max_k \left\{ \phi(x_t, k)^\top \hat{\theta}_t + c_t \sqrt{\phi(x_t, k)^\top A_t^{-1} \phi(x_t, k)} \right\}$

More detail on the combined linear model

For $t = 0 \to T-1$

1. $\forall\, k$, define $A_t = \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda I$ and $\hat{\theta}_t = A_t^{-1} \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau) r_\tau$

2. Observe x_t & choose $a_t = \arg\max_k \left\{ \phi(x_t, k)^\top \hat{\theta}_t + c_t \sqrt{\phi(x_t, k)^\top A_t^{-1} \phi(x_t, k)} \right\}$

3. Observe reward $r_t \sim \nu^{(a_t)}(x_t)$

More detail on the combined linear model

For $t = 0 \to T-1$

1. $\forall k$, define $A_t = \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda I$ and $\hat{\theta}_t = A_t^{-1} \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau) r_\tau$

2. Observe x_t & choose $a_t = \arg\max_k \left\{ \phi(x_t, k)^\top \hat{\theta}_t + c_t \sqrt{\phi(x_t, k)^\top A_t^{-1} \phi(x_t, k)} \right\}$

3. Observe reward $r_t \sim \nu^{(a_t)}(x_t)$

Comments:

More detail on the combined linear model

For $t = 0 \to T-1$

1. $\forall\, k$, define $A_t = \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda I$ and $\hat{\theta}_t = A_t^{-1} \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau) r_\tau$
2. Observe x_t & choose $a_t = \arg\max_k \left\{ \phi(x_t, k)^\top \hat{\theta}_t + c_t \sqrt{\phi(x_t, k)^\top A_t^{-1} \phi(x_t, k)} \right\}$
3. Observe reward $r_t \sim \nu^{(a_t)}(x_t)$

Comments:

i. There is only one A_t and $\hat{\theta}_t$ (not one per arm), so more info shared across k

More detail on the combined linear model

For $t = 0 \to T - 1$

1. $\forall k$, define $A_t = \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda I$ and $\hat{\theta}_t = A_t^{-1} \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau) r_\tau$
2. Observe x_t & choose $a_t = \arg\max_k \left\{ \phi(x_t, k)^\top \hat{\theta}_t + c_t \sqrt{\phi(x_t, k)^\top A_t^{-1} \phi(x_t, k)} \right\}$
3. Observe reward $r_t \sim \nu^{(a_t)}(x_t)$

Comments:

i. There is only one A_t and $\hat{\theta}_t$ (not one per arm), so more info shared across k

ii. Good for large K, but step 2's argmax may be hard

More detail on the combined linear model

For $t = 0 \to T-1$

1. $\forall\, k$, define $A_t = \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau)\phi(x_\tau, a_\tau)^\top + \lambda I$ and $\hat{\theta}_t = A_t^{-1} \sum_{\tau=0}^{t-1} \phi(x_\tau, a_\tau) r_\tau$
2. Observe x_t & choose $a_t = \arg\max_k \left\{ \phi(x_t, k)^\top \hat{\theta}_t + c_t \sqrt{\phi(x_t, k)^\top A_t^{-1} \phi(x_t, k)} \right\}$
3. Observe reward $r_t \sim \nu^{(a_t)}(x_t)$

Comments:

i. There is only one A_t and $\hat{\theta}_t$ (not one per arm), so more info shared across k

ii. Good for large K, but step 2's argmax may be hard

iii. The other formulation, with separate $A_t^{(k)}$ and $\hat{\theta}_t^{(k)}$, is called disjointed

Continuous bandit action spaces

Continuous bandit action spaces

In bandits / contextual bandits, we have always treated the action space as discrete

Continuous bandit action spaces

In bandits / contextual bandits, we have always treated the action space as discrete

This is because we to some extent treated each arm separately, necessitating trying each arm at least a fixed number of times before real learning could begin

Continuous bandit action spaces

In bandits / contextual bandits, we have always treated the action space as discrete

This is because we to some extent treated each arm separately, necessitating trying each arm at least a fixed number of times before real learning could begin

But now with the new combined formulation, there is sufficient sharing across actions that we can learn $\hat{\theta}_t$ and its UCB *without* sampling all arms

Continuous bandit action spaces

In bandits / contextual bandits, we have always treated the action space as discrete

This is because we to some extent treated each arm separately, necessitating trying each arm at least a fixed number of times before real learning could begin

But now with the new combined formulation, there is sufficient sharing across actions that we can learn $\hat{\theta}_t$ and its UCB *without* sampling all arms

This means that in principle, we can now consider continuous action spaces!

Continuous bandit action spaces

In bandits / contextual bandits, we have always treated the action space as discrete

This is because we to some extent treated each arm separately, necessitating trying each arm at least a fixed number of times before real learning could begin

But now with the new combined formulation, there is sufficient sharing across actions that we can learn $\hat{\theta}_t$ and its UCB *without* sampling all arms

This means that in principle, we can now consider continuous action spaces!

This is the power of having a strong model for $\mathbb{E}_{r \sim \nu^{(a_t)}(x_t)}[r]$, and a neural network would serve a similar purpose in place of the combined linear model (UQ less clear)

Continuous bandit action spaces

In bandits / contextual bandits, we have always treated the action space as discrete

This is because we to some extent treated each arm separately, necessitating trying each arm at least a fixed number of times before real learning could begin

But now with the new combined formulation, there is sufficient sharing across actions that we can learn $\hat{\theta}_t$ and its UCB *without* sampling all arms

This means that in principle, we can now consider continuous action spaces!

This is the power of having a strong model for $\mathbb{E}_{r \sim \nu^{(a_t)}(x_t)}[r]$, and a neural network would serve a similar purpose in place of the combined linear model (UQ less clear)

But in principle, there is no “free lunch”, i.e., the hardness of the problem now transfers over to choosing a good model (a bad model will lead to bad performance)