Statistical Learning in Operations Management

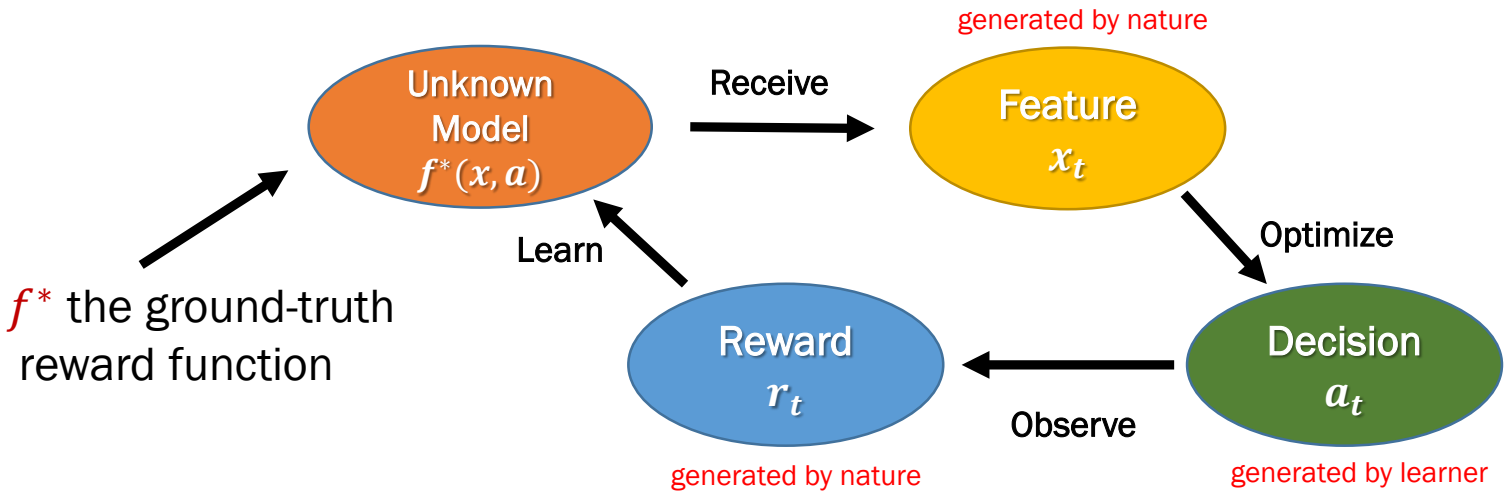
David Simchi-Levi

Executive Summary

- **Strategic Intent:** Develop solutions to leading edge problems for Lab partners through research that brings together data, modeling and analysis to achieve industry leading improvements in business performance.
- **Cross Industry:** Oil/Gas, Retail, Financial Services, Government, Insurance, Airlines, Industrial Equipment, Software
- **Global footprint:** NA, EU, Asia, LA

Online Learning

- No data is available at the beginning of the process.
- Data is generated on-the-fly according to some unknown model and the decisions made by the platform.



- **Objective:** Design algorithms that maximize the accumulated reward, i.e., achieve low regret.
  - Regret = optimal accumulated reward of a clairvoyant – collected accumulated reward

Offline Learning

- The entire data set (of i.i.d. samples) is available at the beginning.
- The decision maker cannot adapt decisions to the new data.

Training data set (i.i.d.)
$(x_1, a_1; r_1), (x_2, a_2; r_2), \cdots, (x_n, a_n; r_n) \sim D$

Offline Learning Algorithms

Predictive functions
$\hat{f}(x, a)$

- **Objective**: Design algorithms that with limited data will generate the $\hat{f}$ so that with high probability $\hat{f}$ will have low error compared to the ground truth f^*.
  - Estimation error = $\mathbb{E}_{(x,a;r)\sim D}[\ell(\hat{f}(x,a), f^*(x,a))]$; it is MSE when ℓ is square loss which is the average squared difference between the estimated values and the actual value

The Interplay between Online and Offline Learning

- Reducing Online Learning to Offline Learning
  - **D. Simchi-Levi and Y. Xu (2020). Bypassing the Monster: A Faster and Simpler Optimal Algorithm for Contextual Bandits under Realizability.**

- Online Learning with Offline Data
  - **J. Bu, D. Simchi-Levi, and Y. Xu (2019). Online Pricing with Offline Data: Phase Transition and Inverse Square Law.**

The Interplay between Online and Offline Learning

- Reducing Online Learning to Offline Learning
  - **D. Simchi-Levi and Y. Xu (2020). Bypassing the Monster: A Faster and Simpler Optimal Algorithm for Contextual Bandits under Realizability.**

- Online Learning with Offline Data
  - **J. Bu, D. Simchi-Levi, and Y. Xu (2019). Online Pricing with Offline Data: Phase Transition and Inverse Square Law.**

Part I: Talk Outline

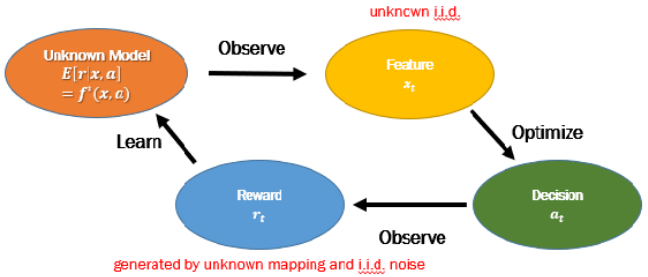
- **Motivation and Research Question**
- **Technical Hurdles and Our Contribution**
- **The Algorithm and Theory**
- **Computational Experiments**

A General Contextual Bandit Model

- For round $t = 1, \cdots, T$
  - Nature generates a random context x_t according to a fixed unknown distribution D_X
  - Learner observes x_t and makes a decision $a_t \in \{1, \ldots, K\}$
  - Nature generates a random reward $r_t(a_t) \in [0,1]$ according to an unknown distribution with conditional mean

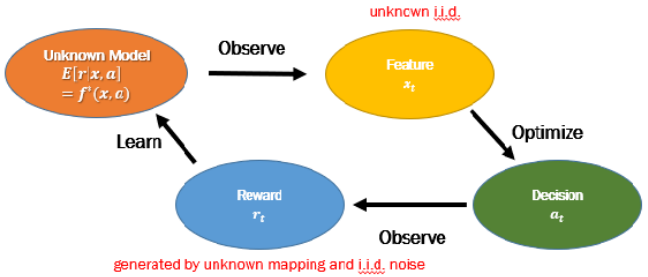
$$\mathbb{E}[r_t(a_t)|x_t = x, a_t = a] = f^*(x, a)$$

- We call f^* the ground-truth reward function; $f^* \in F$
- *Regret*: the total reward loss compared with a clairvoyant who knows f^*
- In statistical learning, people use a function class F to approximate f^*. Examples of F:
  - Linear class / high-dimension linear class / generalized linear models
  - Non-parametric class / reproducing kernel Hilbert space (RKHS)
  - Regression trees
  - Neural networks



Why is the problem important and challenging?

- Contextual bandits combine statistical learning and decision making under uncertainty
- Contextual bandits capture two essential features of sequential decision making under uncertainty
  - Bandit feedback: for each context x_t, learner only observes the reward for her chosen action a_t; no other rewards are observed
    - Learner faces a trade-off between exploration and exploitation
  - Heterogeneity: the effectiveness of each action depends on the context
    - The context space is huge --- Not clear how to learn across contexts for general function class



Literature on Contextual Bandits

- Algorithms:
  - **Upper Confidence Bounds** (Filippi et al. 2010, Rigollet and Zeevi 2010, Abbasi-Yadkori et al. 2011, Chu et al. 2011, Li et al. 2017, …)
  - **Thompson Samplings** (Agrawal and Goyal 2013, Russo et al. 2018, …)
  - **Exponential Weighting** (Auer et al. 2002, McMahan and Streeter 2009, Beygelzimer et al. 2011, …)
  - **Oracle-based** (Dudik et al. 2011, Agarwal et al. 2014, Foster et al. 2018, Foster and Rakhlin 2020, …)
  - **Many Others …**
- Applications:
  - **Recommender systems** (Li et al. 2010, Agarwal et al. 2016, …)
  - **Ride-hailing platforms** (Chen et al. 2019, …)
  - **Dynamic pricing** (Ferreira et al. 2018…)
  - **Healthcare** (Tewari and Murphy 2017, Bastani and Bayati 2020, …)

Relevance to Operations

- Product recommendation:
  - K products
  - T customers arriving in a sequential manner. Each customer has a feature x_t describing gender, age, shopping history, device type, etc.
  - The task it to recommend a product a_t (based on x_t) that generates revenue as high as possible
  - The revenue distribution is unknown, with its conditional mean $f^*(x_t, a_t)$ to be learned
- Personalized medicine:
  - K treatments / dose levels
  - T patients arriving in a sequential manner. Each patient has a feature x_t describing her demographics, diagnosis, genes, etc.
  - The task is to pick a personalized treatment (or dose level) a_t (based on x_t) that is as effective as possible
  - The efficacy is random and unknown, with the efficacy rate $f^*(x_t, a_t)$ to be learned

The Challenge

- We are interested in contextual bandits with a general function class F
- Realizability assumption:

$$f^* \in F$$

- **Statistical challenge**: How can we achieve the optimal regret for any general function class?
- **Computational challenge**: How can we make the algorithm computationally efficient?

- Classical contextual bandits approaches fail to simultaneously address the above two challenges in practice, as they typically
  - Become statistically suboptimal for general F (e.g., UCB variants and Thompson Sampling)
  - Become computationally intractable for large F (e.g., Exponential weighting, Elimination-based methods)

Research Question

- Observation: Given a general function class F, the statistical and computational aspects of "offline regression" are well-studied in ML.
- Specifically, given i.i.d. offline data, advances in ML enable us to find a predictor $\hat{f}$ such that
  - (statistically) $\hat{f}$ achieves low estimation error: support vector machines, random forests, boosting, neural net …
  - (computationally) $\hat{f}$ can be efficiently computed: gradient descent methods
- Can we reduce general contextual bandits to general offline regression?
- **Given** F, and an offline regression oracle, e.g., a least-squares regression oracle

$$\arg\min_{f \in F} \sum_{t=1}^{n} (f(x_t, a_t) - r_t(a_t))^2$$

or its regularized counterparts (e.g., Ridge and Lasso),

Challenge: Design a contextual bandit algorithm such that

- (statistically) it achieves the optimal regret whenever the offline regression oracle attains the optimal estimation error
- (computationally) it requires no more computation than calling the offline regression oracle

- An open problem mentioned in Agarwal et al. (2012), Foster et al. (2018), Foster and Rakhlin (2020)

Talk Outline

- **Motivation and Research Question**
- **Technical Hurdles and Our Contribution**
- **The Algorithm and Theory**
- **Computational Experiments**

Why is the research question so challenging?

- Two key challenges for reducing contextual bandits to offline regression
  - 1. Statistical difficulties associated with confidence bounds
  - 2. Statistical difficulties associated with analyzing dependent actions

1. Stat. difficulties with confidence bounds

- Many classical contextual bandits algorithms, e.g., UCB and Thompson Sampling, only work with certain parametric models
- This is because they usually rely on effective confidence bounds constructed for each (x, a) pair
- While this is possible for a simple class F, like the linear class, it is impossible for a general F
- Foster et al. (2018) propose a computationally efficient confidence-bounds-based algorithm using an offline regression oracle
  - The algorithm only has statistical guarantees under some strong distributional assumptions

2. Stat. difficulties with analyzing dependent actions

- Translating offline estimation error guarantees to contextual bandits is a challenge
- This is because the data collected in the learning process is not i.i.d
  - The action distribution in later rounds depend on the data in previous rounds
- Recently, Foster and Rakhin (2020) develop an optimal and efficient algorithm for contextual bandits assuming access to an online regression oracle
- The online regression oracle provides statistical guarantees for an arbitrary data sequence possibly generated by an (adaptive) adversary
- Computationally efficient algorithms for the required online regression oracle are only known for specific function classes
  - Lack of efficient algorithms for many natural function classes, e.g., sparse linear class, Hölder classes, neural networks, ...

Our Contribution

- We provide the first optimal and efficient **black-box reduction** from general contextual bandits to offline regression
  - The algorithm is simpler and faster than existing approaches to general contextual bandits
    - The design of the algorithm builds on Abe and Long (1999), Agrawal et al. (2014), Foster and Rakhlin (2020)
    - The analysis of the algorithm is highly non-trivial and reveals surprising connections between several historical approaches to contextual bandits
  - Any advances in offline regression immediately translate to contextual bandits, statistically and computationally

Our Contribution

- Our algorithm's computational complexity is much better than existing algorithms for complicated F

Table 1 **Algorithms' performance with general finite $\mathcal{F}$. Advantages are marked in bold.**

Algorithm	Statistical optimality	Computational complexity
`Regressor Elimination` (Agarwal et al. 2012)	**optimal**	$\Omega(\lvert\mathcal{F}\rvert)$ intractable
`ILOVETOCONBANDITS` (Agarwal et al. 2014)	**optimal**	$\widetilde{O}(\sqrt{KT/\log\lvert\mathcal{F}\rvert})$ calls to an offline classification oracle
`RegCB` (Foster et al. 2018)	suboptimal	$O(T^{3/2})$ calls to an **offline regression** oracle
`SquareCB` (Foster and Rakhlin 2020)	**optimal**	$O(T)$ calls to an online regression oracle
`FALCON` (this paper)	**optimal**	$\boldsymbol{O(\log T)}$ or $\boldsymbol{O(\log\log T)}$ calls to an **offline regression** oracle

FALCON's computational complexity is equivalent to solving a few offline regression problems

What Does "Monster" Refer To?

- In the contextual bandit literature, "monster" refers to algorithms that require huge amount of computation
- Dudik M, Hsu D, Kale S, Karampatziakis N, Langford J, Reyzin L, Zhang T (2011) Efficient optimal learning for contextual bandits.
  - These authors refer to their paper as the "Monster Paper"
  - Optimal regret but requires "a monster amount of computation"
- Agarwal A, Hsu D, Kale S, Langford J, Li L, Schapire R (2014) Taming the Monster: A fast and simple algorithm for contextual bandits.
  - Optimal regret with reduced computational cost
  - Requires using the offline classification oracle
- This paper: Bypassing the Monster
  - Under a weak "realizability" assumption: $f^* \in F$

Talk Outline

- **Motivation and Research Question**
- **Technical Hurdles and Our Contribution**
- **The Algorithm and Theory**
- **Computational Experiments**

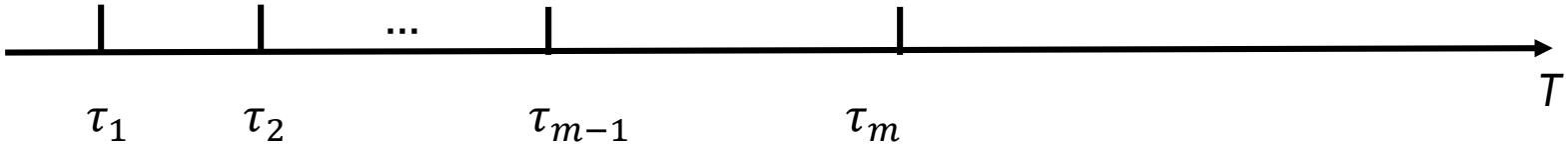
The algorithm

- Three components
  - An epoch schedule (to exponentially save computation)
  - Greedy calls to the offline regression oracle (to obtain reward predictor)
  - A sampling rule (=randomized algorithm over actions) determined by the predictor and an epoch-varying learning rate (to make decisions)
    - The sampling rule is introduced by Abe and Long (1999) and adopted in Foster and Rakhlin (2020)
- The algorithm is fast and we call it **FALCON** (FAst Least-squares-regression-oracle for CONtextual bandits)

Falcon, the fastest animal on earth
Source: Kirstin Fawcett, fakuto.com

Component 1: Epoch Schedule

- An epoch schedule $\tau_1, \tau_2, \ldots$



- The algorithm only calls the regression oracle at the start of each epoch.
  - When $\tau_m = 2^m$, it only makes $O(\log T)$ calls to the oracle over T rounds
  - When T is known, the oracle calls can be reduced to $O(\log \log T)$ (which is not a trivial property and is useful in clinical trials)
- This implies that the oracle is called less and less frequently as the algorithm proceeds

Component 2: Oracle Calls

- Before the start of each epoch m, solves

$$\min_{f \in F} \sum_{t=1}^{\tau_{m-1}} (f(x_t, a_t) - r_t(x_t, a_t))^2$$

via the least square oracle, and obtains a predictor $\hat{f}_m$

- We can replace the least squares oracle by any other offline regression oracles (e.g., regularized oracles like *Ridge* and *Lasso*)
- What to do next for making decisions?
- If we directly follow the predictor to choose greedy actions, then the algorithm does not explore at all, and may perform poor
  - We address the exploration-exploitation dilemma via sampling

Component 3: Sampling Rule

The learning rate balances between exploration and exploitation. The algorithm explores more at the beginning and gradually exploits more.

- For each epoch m, we have a learning rate $\gamma_m \asymp \sqrt{\tau_{m-1}}$
- At round t, we do the following

Observe context $x_t \in \mathcal{X}$.

Compute $\widehat{f}_m(x_t, a)$ for each action $a \in \mathcal{A}$. Let $\widehat{a}_t = \max_{a \in \mathcal{A}} \widehat{f}_m(x_t, a)$. Define

The greedy action

$$p_t(a) = \begin{cases} \frac{1}{K + \gamma_m \left(\widehat{f}_m(x_t, \widehat{a}_t) - \widehat{f}_m(x_t, a) \right)}, & \text{for all } a \neq \widehat{a}_t, \\ 1 - \sum_{a \neq \widehat{a}_t} p_t(a), & \text{for } a = \widehat{a}_t. \end{cases}$$

Sample $a_t \sim p_t(\cdot)$ and observe reward $r_t(a_t)$.

The probability of selecting the greedy action is the highest. This corresponds to "exploitation"

The probability of selecting each non-greedy action is inversely proportional to the predicted gap between this action and the greedy action, as well as the learning rate γ_m. This corresponds to "exploration"

Statistical Guarantees: Finite Function Class

- **Theorem**: FALCON guarantees expected regret of

$$\tilde{O}\left(\sqrt{KT\log|F|}\right)$$

through $O(\log T)$ calls to the least-squares regression oracle. The number of oracle calls can be reduced to $O(\log\log T)$ if T is known in advance.

- Combined with a $\Omega\left(\sqrt{KT\log|F|}\right)$ lower bound (Agrawal et al. 2012), we know that our regret is minimax optimal.

Statistical Guarantees: General Function Class

- **Input (offline guarantee)**: Given n i.i.d. samples $(x_i, a_i; r_i) \sim D$, and an offline regression oracle that returns an estimator $\hat{f}$ such that $\forall$ possible D,

$$\mathbb{E}_{(x,a;r)\sim D}\left[\left(\hat{f}(x,a) - f^*(x,a)\right)^2\right] \leq \boldsymbol{Er(n;F)}$$

where the estimation error guarantee $\boldsymbol{Er(n;F)}$ depends on the number of samples n and the complexity of F

- **Theorem**: Given an offline regression oracle with estimation error $Er(n;F)$ for n samples, FALCON guarantees expected regret of

$$\tilde{O}\left(\sqrt{KEr(T;F)T}\right)$$

through $O(\log T)$ calls to the offline regression oracle. The number of oracle calls can be reduced to $O(\log\log T)$ if T is known

  - Plugging in the rate-optimal $Er(n;F)$ ensures that the regret is optimal in terms of T, which matches the regret lower bound proved in Foster and Rakhlin (2020).

Examples

- When F is a linear class with dimension d
  - Least squares estimator ensures $Er(n; F) = O\left(\frac{d}{n}\right)$
  - FALCON achieves $O(\sqrt{KT(d + \log T)})$ regret using the least squares regression oracle. While the dependence on K is suboptimal, the dependence on T improves over best known algorithm by a $\log T$ factor

- When F is a linear class with sparsity s
  - LASSO ensures $Er(n; F) = \tilde{O}\left(\frac{s\log d}{n}\right)$ (under certain conditions of D_X)
  - FALCON achieves $\tilde{O}(\sqrt{Ks\,T\log d})$ regret using LASSO as the offline oracle

- When F is a class of neural network
  - There are many methods to find an estimator $\hat{f}$ that perform extremely well in practice
  - Our results can directly transform low estimation error into the best-possible regret bound (based on such error), theoretically or empirically

- Other Examples: generalized linear models, non-parametric classes, …

Proof Sketch

- Translating offline estimation error guarantees to contextual bandits is a challenge
- Data collected in the online learning process is not i.i.d
- Offline guarantees provide upper bounds on the "distance" between $\hat{f}$ and f^* for a fixed action distribution

- **Initialization.** A dual interpretation: our algorithm adaptively maintains a distribution over policies in the *universal policy space* $\Psi = [K]^X$
  - A policy $\pi: X \mapsto [K]$ is a deterministic decision function
    - Let π_{f^*} be the true optimal policy, and $\pi_{\hat{f}_m}$ be the greedy policy
  - At epoch m, $\hat{f}_m$ and γ_m induce a distribution over policies $Q_m(\cdot)$
    - $Q_m(\pi) = \prod_{x \in X} p_m(\pi(x)|x)$, where $p_m(\pi(x)|x)$ is the probability that the sampling rule selects action $\pi(x)$ given context x

Proof Sketch

Our algorithm's (per-round) expected regret if $\hat{f}_m$ were ground-truth

Step 1. Per-round property: At each epoch m, given $\hat{f}_m$ and γ_m, the distribution Q_m ensures that

$$\sum_{\pi \in \Psi} Q_m(\pi) \mathbb{E}_{D_X}\left[\hat{f}_m\left(x, \pi_{\hat{f}_m}(x)\right) - \hat{f}_m\left(x, \pi(x)\right)\right] = O(K/\gamma_m)$$

Estimated per-round expected regret of π

This bound is directly guaranteed by the sampling rule

Step 2. Proof by induction: At each epoch m, $Q_1, \ldots, Q_{m-1}$ ensure that for all $\pi \in \Psi$,

$$\mathbb{E}_{D_X}\left[f^*\left(x, \pi_{f^*}(x)\right) - f^*(x, \pi(x))\right] \leq 2\mathbb{E}_{D_X}\left[\hat{f}_m\left(x, \pi_{\hat{f}_m}(x)\right) - \hat{f}_m\left(x, \pi(x)\right)\right] + O(K/\gamma_m)$$

True per-round expected regret of π

Step 3. Putting together: At each epoch m, our per-round expected regret is

$$\sum_{\pi \in \Psi} Q_m(\pi) \mathbb{E}_{D_X}\left[f^*\left(x, \pi_{f^*}(x)\right) - f^*(x, \pi(x))\right]$$

by step 2

$$\leq 2\sum_{\pi \in \Psi} Q_m(\pi) \mathbb{E}_{D_X}\left[\hat{f}_m\left(x, \pi_{\hat{f}_m}(x)\right) - \hat{f}_m\left(x, \pi(x)\right)\right] + O(K/\gamma_m)$$

by step 1

$$= O(K/\gamma_m)$$

We choose $\{\gamma_m\}$ such that Step 2 holds and Step 3 leads to the optimal accumulated regret

A closer look at Step 2

Step 2. Proof by induction: At each epoch m, $Q_1, \ldots, Q_{m-1}$ ensure that for all $\pi \in \Psi$,

$$\mathbb{E}_{D_X}\left[f^*\left(x, \pi_{f^*}(x)\right) - f^*(x, \pi(x))\right] \leq 2\mathbb{E}_{D_X}\left[\hat{f}_m\left(x, \pi_{\hat{f}_m}(x)\right) - \hat{f}_m\left(x, \pi(x)\right)\right] + O(K/\gamma_m)$$

True per-round expected regret of π

Estimated per-round expected regret of π

- It connects $\hat{f}_m$ and f^* without specifying an action distribution
  - It connects the "estimated world" and the "true world"
- It holds for non-iid and dependent decision process (of $\{a_t\}$)
  - The induction argument shows how exploration in early rounds benefit exploitation in later rounds
- It utilizes the iid properties of $\{x_t\}$
- It establishes a bridge from offline estimation guarantees to online decision making guarantees
  - The analysis is general and does not rely on any refined property of F

A Few Observations

- The Statistical Guarantees hold even if the MSE Loss Function is replaced by Strongly Convex Loss Function
  - Generalized Linear Model: Logistics Loss Function

- Comparing FALCON to SquareCB (Foster and Rakhlin, 2020)
  - FALCON assumes iid contexts; SquareCB allows for general contexts
  - Computational Efficient Algorithms:
    - SquareCB requires computationally efficient algorithms for online regression oracle. This is only known for specific function classes.
    - Many more functions classes are covered by computational efficient offline regression oracles (as required by FALCON)
  - FALCON allows occasional updates; SquareCB requires continuous updates
    - Important in healthcare applications where rewards are delayed

Talk Outline

- **Motivation and Research Question**
- **Technical Hurdles and Our Contribution**
- **The Algorithm and Theory**
- **Computational Experiments**

Initial Computational Experiments

On real-world datasets for three types of problems:

- Multiclass Classification,
- Recommendation,
- Price Optimization.

Classification & Recommendation Datasets

- 10 multiclass classification data sets from OpenML and 2 learning-to-rank data sets from Microsoft and Yahoo! Public Data Set Website.

- Reduction to Online Contextual Bandit Problems.
  - 0/1 Loss encoding

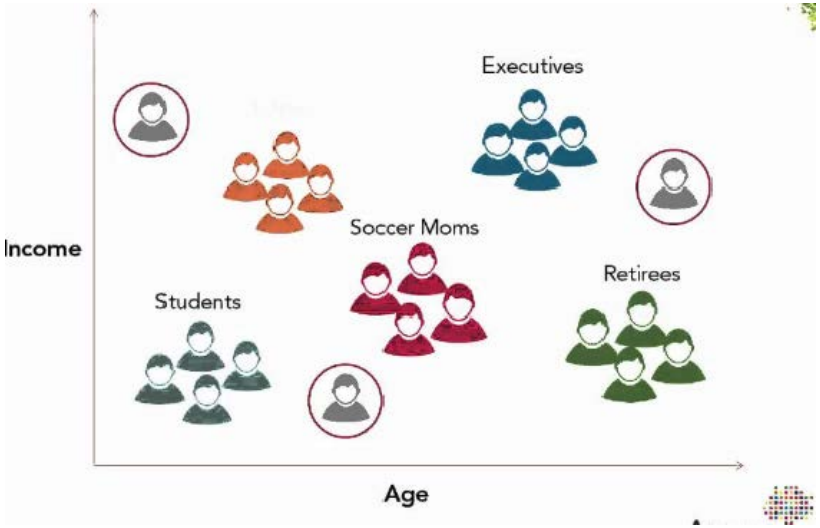
Retail Applications: Classification

- Multi-class classification is a fundamental task in ML. Online multi-class classification can be used for many applications from handwriting recognition and face recognition to customer group recognition and promotion design.
- Classify the customer correctly, then we incur 0 loss; otherwise we incur a loss of 1.

handwriting recognition

face recognition

emotion recognition



Retailer Applications: Recommendation

- A recommendation system seeks to predict the ranking (or rating) that a user would give to a product.
- When a customer arrives, we want to recommend a product that she likes most. If we recommend a product she ranked high, then we incur smaller loss; otherwise we incur larger loss.

Benchmark Algorithms

- For simplicity, we use benchmark contextual bandit algorithms implemented in Vowpal Wabbit (v8.8.0), an open-source library for online learning algorithms
  - Greedy: use only greedy action
  - ϵ-Greedy: use greedy action with prob (1- ϵ) and uniform on all other actions
  - Online Cover & Cover NU: heuristic versions of "Taming the Monster," see Agarwal et al. 2014
  - Bagging & Bagging Greedy: heuristic versions of Thompson Sampling
  - RegCB-elimination: a generalization of "successive elimination," see Foster et al. 2018
  - RegCB-optimistic: a generalization of UCB, see Foster et al. 2018

Stat significant win-loss difference across 12 datasets

The statistical significance is defined based on approximate Z-test
FALCON 1 uses a linear class with least squares estimator
FALCON 2 uses a linear class with ridge estimator
FALCON 3 uses a regression tree class with gradient boosting estimator

Table 4: Statistically Significant win-loss difference of row against column for ranking + openml datasets

↓ vs →	greedy	$\epsilon-$greedy	cover	cover-nu	bag	bag-greedy	regcb-e	regcb-o	FALCON1	FALCON2	FALCON3
greedy	-	-1	-1	-2	0	-4	-3	-5	-3	-4	-5
$\epsilon-$greedy	1	-	0	-2	-1	-1	-1	-6	-2	-3	-4
cover	1	0	-	0	1	0	-1	-6	-2	-3	-6
cover-nu	2	2	0	-	-1	-1	-3	-5	-4	-4	-4
bag	0	1	-1	1	-	-2	1	-6	-1	-2	-4
bag-greedy	4	1	0	1	2	-	-1	-5	0	-1	-2
regcb-e	3	1	1	3	-1	1	-	-2	1	1	-3
regcb-o	5	6	6	5	6	5	2	-	0	-1	-2
FALCON1	3	2	2	4	1	0	-1	0	-	-4	-3
FALCON2	4	3	3	4	2	1	-1	1	4	-	-3
FALCON3	5	4	6	4	4	2	3	2	3	3	-

Stat significant win-loss difference across 12 datasets

The statistical significance is defined based on approximate Z-test
FALCON 1 uses a linear class with least squares estimator
FALCON 2 uses a linear class with ridge estimator
FALCON 3 uses a regression tree class with gradient boosting estimator

Table 4: Statistically Significant win-loss difference of row against column for ranking + openml datasets

↓ vs →	greedy	$\epsilon-$greedy	cover	cover-nu	bag	bag-greedy	regcb-e	regcb-o	FALCON1	FALCON2	FALCON3
greedy	-	-1	-1	-2	0	-4	-3	-5	-3	-4	-5
$\epsilon-$greedy	1	-	0	-2	-1	-1	-1	-6	-2	-3	-4
cover	1	0	-	0	1	0	-1	-6	-2	-3	-6
cover-nu	2	2	0	-	-1	-1	-3	-5	-4	-4	-4
bag	0	1	-1	1	-	-2	1	-6	-1	-2	-4
bag-greedy	4	1	0	1	2	-	-1	-5	0	-1	-2
regcb-e	3	1	1	3	-1	1	-	-2	1	1	-3
regcb-o	5	6	6	5	6	5	2	-	0	-1	-2
FALCON1	3	2	2	4	1	0	-1	0	-	-4	-3
FALCON2	4	3	3	4	2	1	-1	1	4	-	-3
FALCON3	5	4	6	4	4	2	3	2	3	3	-

Each entry shows the statistically significant win-loss of a raw against a column

Stat significant win-loss difference across 12 datasets

The statistical significance is defined based on approximate Z-test
FALCON 1 uses a linear class with least squares estimator
FALCON 2 uses a linear class with ridge estimator
FALCON 3 uses a regression tree class with gradient boosting estimator

Table 4: Statistically Significant win-loss difference of row against column for ranking + openml datasets

↓ vs →	greedy	$\epsilon-$greedy	cover	cover-nu	bag	bag-greedy	regcb-e	regcb-o	FALCON1	FALCON2	FALCON3
greedy	-	-1	-1	-2	0	-4	-3	-5	-3	-4	-5
$\epsilon-$greedy	1	-	0	-2	-1	-1	-1	-6	-2	-3	-4
cover	1	0	-	0	1	0	-1	-6	-2	-3	-6
cover-nu	2	2	0	-	-1	-1	-3	-5	-4	-4	-4
bag	0	1	-1	1	-	-2	1	-6	-1	-2	-4
bag-greedy	4	1	0	1	2	-	-1	-5	0	-1	-2
regcb-e	3	1	1	3	-1	1	-	-2	1	1	-3
regcb-o	5	6	6	5	6	5	2	-	0	-1	-2
FALCON1	3	2	2	4	1	0	-1	0	-	-4	-3
FALCON2	4	3	3	4	2	1	-1	1	4	-	-3
FALCON3	5	4	6	4	4	2	3	2	3	3	-

Stat significant win-loss difference across 12 datasets

The statistical significance is defined based on approximate Z-test
FALCON 1 uses a linear class with least squares estimator
FALCON 2 uses a linear class with ridge estimator
FALCON 3 uses a regression tree class with gradient boosting estimator

Table 4: Statistically Significant win-loss difference of row against column for ranking + openml datasets

↓ vs →	greedy	$\epsilon-$greedy	cover	cover-nu	bag	bag-greedy	regcb-e	regcb-o	FALCON1	FALCON2	FALCON3
greedy	-	-1	-1	-2	0	-4	-3	-5	-3	-4	-5
$\epsilon-$greedy	1	-	0	-2	-1	-1	-1	-6	-2	-3	-4
cover	1	0	-	0	1	0	-1	-6	-2	-3	-6
cover-nu	2	2	0	-	-1	-1	-3	-5	-4	-4	-4
bag	0	1	-1	1	-	-2	1	-6	-1	-2	-4
bag-greedy	4	1	0	1	2	-	-1	-5	0	-1	-2
regcb-e	3	1	1	3	-1	1	-	-2	1	1	-3
regcb-o	5	6	6	5	6	5	2	-	0	-1	-2
FALCON1	3	2	2	4	1	0	-1	0	-	-4	-3
FALCON2	4	3	3	4	2	1	-1	1	4	-	-3
FALCON3	5	4	6	4	4	2	3	2	3	3	-

Dynamic Pricing

- The public data set for revenue management is retrieved from CPRM, Columbia University.
- 200,000 examples of Auto Loans across 134 days
- 0/1 response on “apply / not apply”
- 144 features
  - We selected the five most important features: CarType, Primary_FICO, Term, Competition_Rate, OneMonth.
- We use cumulative revenue to evaluate performance of each algorithm in the simulation.

Setting	Price Range	# of Price Choices	Price
1	[500, 7500]	15	$p_i = 500i, i = 1, 2, ..., 15$
2	[1000, 10000]	10	$p_i = 1000i, i = 1, 2, ..., 10$

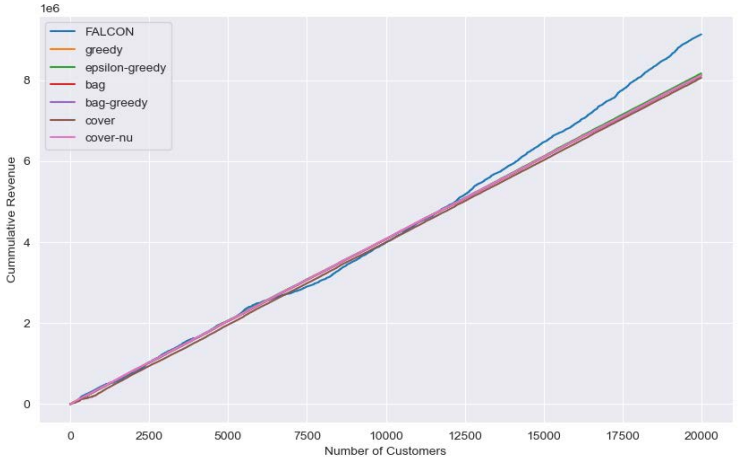
Linear Classes Across Algorithms

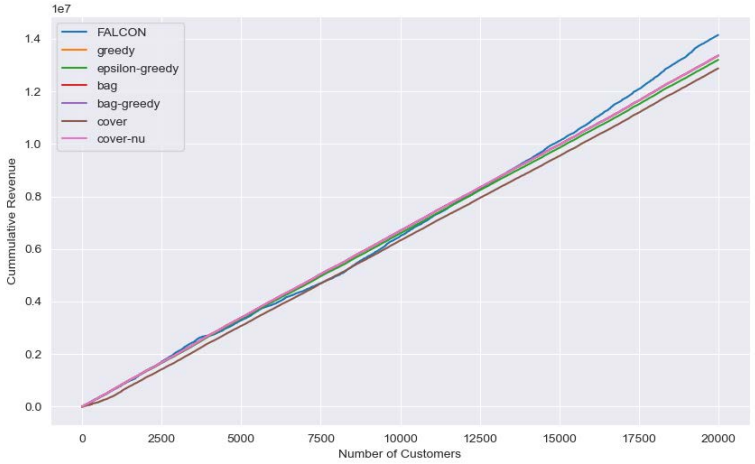
- Here is the revenue performance across algorithms for two settings.

Table 1: Maximum Average Revenues for Online Auto Loan dataset (10%)

Data Set id	greedy	$\epsilon-$greedy	cover	cover-nu	bag	bag-greedy	regcb-e	regcb-o	FALCON
1	405.75	408.375	408.65	406.125	405.75	405.75	412.65	405.75	**442.275**
2	668.05	660.05	653.9	668.05	668.05	668.1	669	668.05	**697.55**

- Below are comparison of different algorithms in Set 1&2:



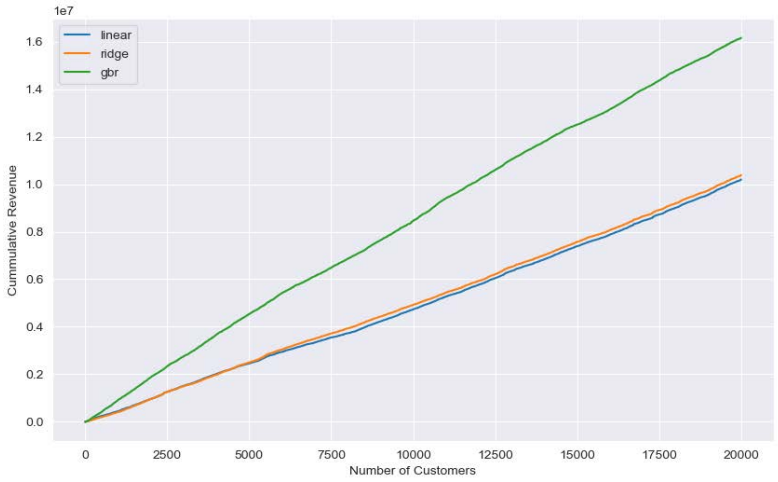


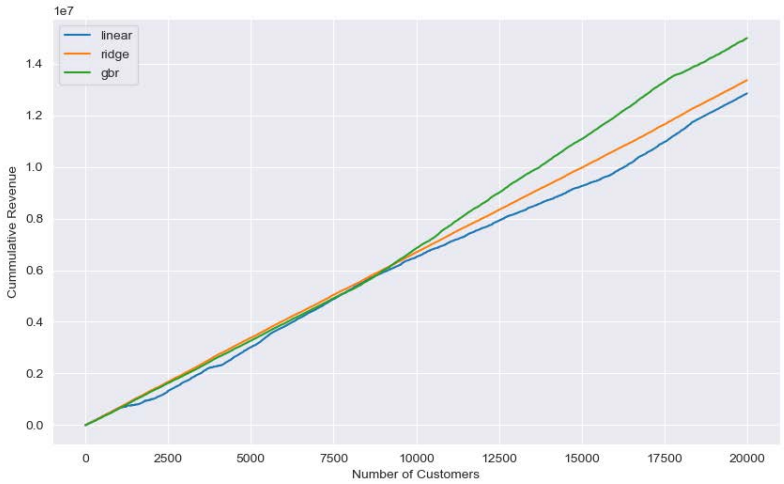
Function Classes in FALCON

- We consider Linear / Ridge / GradientBoosting regressions for these two settings and GBR performs better as it eliminates misspecification.

Table 2: Maximum Average Revenues for Online Auto Loan dataset for `FALCON` (10%)

Data Set id / Function Class	Linear (least squares)	Linear (ridge)	Regression Trees (gbr)
1	442.275	526.7	**819.425**
2	697.55	668.15	**821.75**





References

- D. Simchi-Levi and Y. Xu (2020). Bypassing the Monster: A Faster and Simpler Optimal Algorithm for Contextual Bandits under Realizability.
  - Mathematics of Operations Research, to appear
- D. Foster, A. Rakhlin, D. Simchi-Levi, and Y. Xu (2020). Instance-Dependent Complexity of Contextual Bandits and Reinforcement Learning: A Disagreement-based Perspective
  - Conference version appeared in COLT 2021

Other Extensions

- Xu and Zeevi (2020) extend our results to contextual bandits with infinite actions. They also introduce a new optimism-based algorithmic principle.
- Krishnamurthy et al. (2021) extend our results to the setting where F is misspecified.
- Wei and Luo (2021) extend our results to non-stationary contextual bandits.
- Sen et al. (2021) extend our results to contextual bandits with a combinatorial action space.

Thank you!