Semi-parametric dynamic contextual pricing

Virag Shah — Jose Blanchet — Ramesh Johari
Uber Inc, Stanford University
virag@uber.com

December 5, 2019

Dynamic pricing

- Several e-commerce platforms have access to data describing history of different users and types of different products.
- Platforms can leverage this information for pricing, and optimizing revenue.
- This requires learning online the mapping from user context to optimal price, efficiently.

Distinguishing features of our setting

We believe that the following are important features and ours is the first work to incorporate all of them.

1. *Binary feedback:* Customer buys the item, or she does not. Her true valuation is not known.

2. *Contextual:* Platform needs to learn the relationship between the covariates and the expected valuation.

3. *Non-parametric residuals:* The residual uncertainty in valuation given covariates is assumed non-parametric.

Summary of Related Work

	Contextual	Non-parametric residuals	Binary feedback
Kleinberg and Leighton (2003)		✓	✓
Javanmard and Nazerzadeh (2019)	✓		✓
Qiang and Bayati (2019)	✓	✓	
Cohen et al. (2016b); Mao et al. (2018)	✓		✓
Ban and Keskin (2019)	✓	✓	
	✓		✓
Nambiar et al. (2019)	✓	✓	
Our work	✓	✓	✓

– Look at our NeurIPS 2019 paper for further details.

Basic Framework

- Discrete times $1, 2, \ldots, n$, one user arrives per time step
- Each user is shown one product, which is ex-ante fixed
- Let V_t be the value t^{th} user assigns to the product.
- Let p_t be the price set by the platform.
- The user buys the product if $p_t \leq V_t$.
- Platform does not know or observe V_t, but has access to covariates $X_t \in \mathbb{R}^d$ which may describe user's history and product's type
- Goal: set prices $p_1, \ldots, p_n$ so as to maximize $\sum_{t=1}^{n} p_t \mathbf{1}\{p_t \leq V_t\}$.

The Data available till time t

- Input: $\{X_i, p_i\}_{i=1}^{t-1}$.
  X_t : covariate. p_t: price

- Output: $\{Y_i\}_{i=1}^{t-1}$, where

$$Y_i = \begin{cases} 1 \text{ if } V_i \geq p_i \\ 0 \text{ otherwise} \end{cases} .$$

  In other words, Y_i captures whether i^{th} was success or failure.

The Semi-parametric Model for Valuation

- We let

$$\ln V_i = \theta_0^\intercal X_i + Z_i,$$

  θ_0: unknown parameters, Z_i: unknown residuals/noise.

- Residuals Z_i are i.i.d. with unknown (non-parametric) distribution.

- Covariates X_i i.i.d. with unknown distribution.

Exploration-exploitation tradeoff

- Exploration: Experiment with prices p_t to better learn θ_0 and distribution of noise Z
- Exploitation: Choose price p_t to maximize revenue.
- Recall, the goal is to maximize platform's long term revenue: $\Gamma_n = \sum_{t=1}^{n} p_t \mathbf{1}\{p_t \leq V_t\}$.

The Oracle

- We study regret against the Oracle which knows θ_0 and the distribution of Z.

– *Optimal policy for the* Oracle *:*

- Let $F(z) = z\mathbb{P}(Z \geq \ln z)$, and $z^* = \arg\sup_z F(z)$.
- Here, $F(z)$ would be the revenue function if covariates X_t were 0.

Proposition

The following pricing policy maximizes revenue for the Oracle*: At each time t set price p_t^* such that*

$$\ln p_t^* = \theta_0^\intercal X_t + \ln z^*.$$

Designing Optimal Bandit Algorithm: Key Ideas

- Recall, revenue maximizing policy for Oracle: $\ln p_t^* = \theta_0^\intercal X_t + \ln z^*$.

- For each z and θ, think of (z, θ) as an arm (i.e. a potential option). Pulling arm (z, θ) is equivalent to setting price p_t such that $\ln p_t = \theta^\intercal X_t + \ln z$.

- $(z, \theta) \in \mathbb{R}^{d+1}$: Curse of dimensionality?

- Important observation: Given X_t, for each choice of price p_t we *simultaneously* obtain information about the expected revenue for a *range* of pairs (z, θ).

DEEP-C Pricing Algorithm: Summary

DEEP-C: Dynamic Experimentation and Elimination of Prices - with Covariates.

- Maintain a set $A(t)$ of 'active arms' (z, θ) at each time.

- At time t, observe X_t and compute the set of active prices:

$$P(t) = \{p_t : \exists (z, \theta) \in A(t) \text{ s.t. } \ln p_t = \theta^\intercal X_t + \ln z\}.$$

- Choose price p_t at random from $P(t)$.

- Observe the revenue obtained. Eliminate (z, θ)'s from $A(t)$ for which there is enough information about sub-optimality.

The main result

Under some smoothness, compactness, independence, etc. assumptions, the following holds.

Theorem

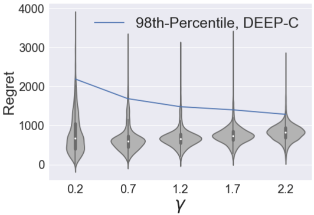
The expected regret satisfies the following: there exists a constant c such that

$$\mathbb{E}[R_n] = O\left(d^c\sqrt{n}\right).$$

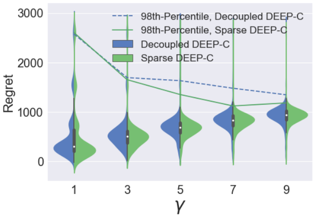
Reducing time-complexity by leveraging sparsity

- DEEP-C resolves sample complexity part, but computational complexity scales poorly with d.
- We assume and leverage sparsity in θ_0 to resolve computational complexity
- **Decoupled DEEP-C:**
  - Phase 1: Choose prices at random. Estimate θ_0 using sparse semi-parametric regression.
  - Phase 2: Use a one-dimensional version of DEEP-C to estimate z^* and maximize revenue.
- **Sparse DEEP-C:** No decoupling. Simultaneously estimate θ_0 using sparse semi-parametric regression and use DEEP-C to estimate z^* and maximize revenue.

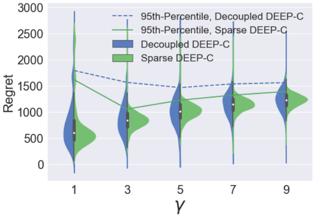
Regret comparison of the policies.



(a) DEEP-C, $d = 2$.



(b) DEEP-C variants, $d = 2$



(c) DEEP-C variants, $d = 100$

Conclusions

- To learn via price experimentation, we do not need to make parametric (probit/logistic/generalized-linear type) assumptions.
- We have a provably efficient algorithm which works under a ‘very general’ setting.