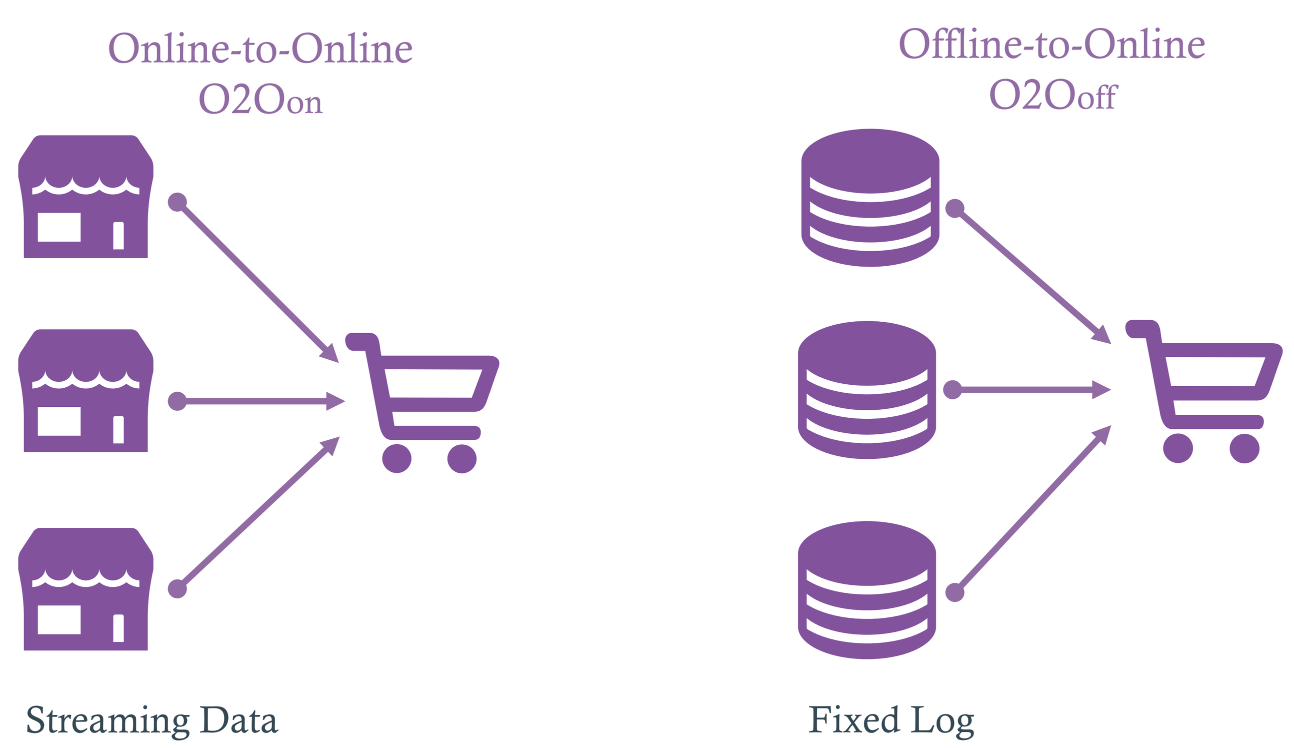
Transfer Faster, Price Smarter: Minimax Dynamic Pricing under Cross-Market Preference Shift

Yi Zhang, Columbia University | Elynn Chen, New York University | Yujun Yan, Dartmouth College

NeurIPS

Motivation

- **Problem**: Dynamic pricing is vital for e-commerce, ride-sharing, airlines.
- **Challenge**: New markets suffer from data scarcity, while mature markets generate abundant logs.
- **Gap**: Existing transfer methods assume identical utilities across markets, failing under preference shift.
- **Goal**: Design a framework to leverage auxiliary markets while provably handling structured model shift.



Cross-Market Transfer Dynamic Pricing

Aggregate → Debias Framework

- The **intuition** behind this framework arises naturally from a structured handling of the **bias-variance tradeoff**.
- **Pooling** all market data yields a **low-variance, high-bias** estimate, as it ignores cross-market differences. Instead of directly relying on this, we treat it as a *rough draft* and perform careful corrections.
- This resembles an editor who adjusts only problematic sentences, preserving valuable shared information.
- **Debiasing** acts as a precision tool that corrects for local market deviations without discarding useful global patterns.

Discussion

- **Key insight:** CM-TDP bridges meta-learning, robust statistics, and bandits to deliver practical transfer pricing.
- **Impact**: Enables faster revenue convergence in new markets while being robust to preference differences.
- **Future work**: (i) Handling covariate shift; (ii) robust detection against adversarial markets; (iii) extending similarity measures beyond sparsity.

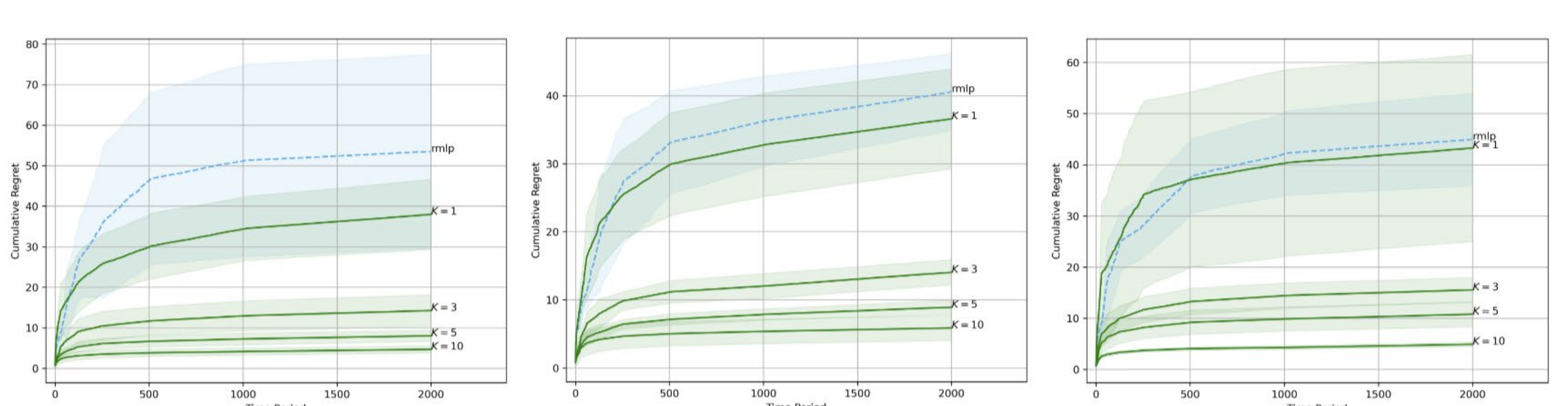
Two Utility Classes

Linear Utility Function

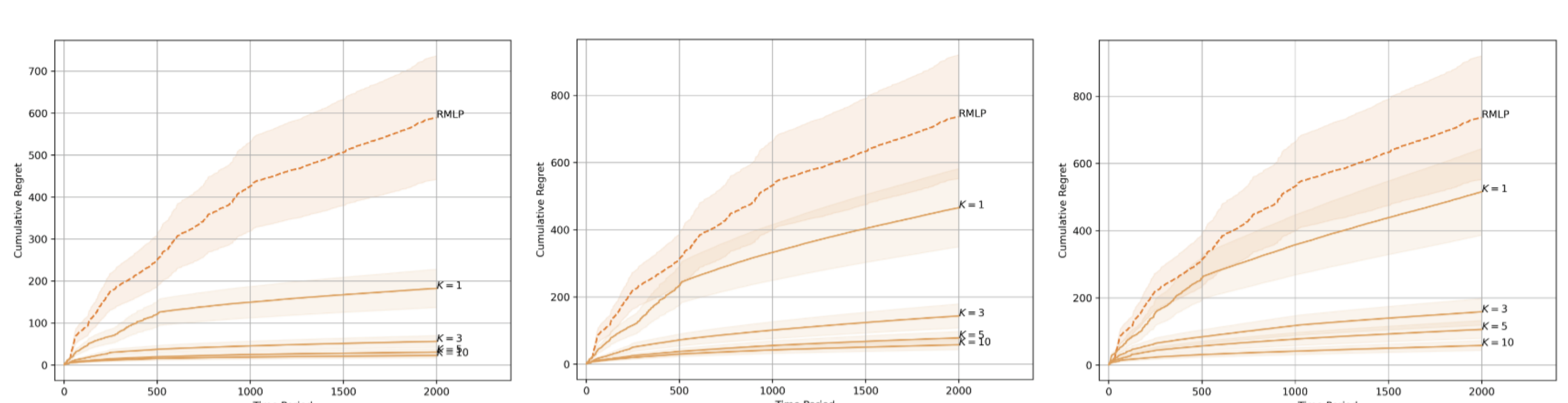
- **Estimation**: Maximum likelihood estimation
- **Utility Function**: $v_t^{(k)} = x_t^{(k)} \cdot \beta^{(k)} + \varepsilon_t$, $k \in \{0\} \cup [K]$
- **Task Similarity**: $\max_{k\in[K]} \|\beta^{(k)} - \beta^{(0)}\| \leq s_0$
- **Regret Upper Bound**: $\mathcal{O}(\frac{d}{K} \cdot \log d \cdot \log T + s_0 \cdot \log d \cdot \log T)$
- **Regret Lower Bound**: $\mathcal{O}(\frac{d}{K} \cdot \log T + s_0 \cdot \log \frac{d}{s_0} \cdot \log T)$

RKHS Utility Function

- **Estimation**: Kernel Logistic Regression
- **Utility Function**: $v_t^{(k)} = g^{(k)}\left(x_t^{(k)}\right) + \varepsilon_t, g^{(k)} \in \mathcal{H}_k$, $k \in \{0\} \cup [K]$
- **Task Similarity**: $\max_{k\in[K]} \|g^{(k)} - g^{(0)}\|_K \leq H$
- **Regret Upper Bound**: $\mathcal{O}(K^{-\frac{2\alpha\beta}{2\alpha\beta+1}} \cdot T^{\frac{1}{2\alpha\beta+1}} + H^{\frac{2}{2\alpha+1}} \cdot T^{\frac{1}{2\alpha+1}})$
- **Regret Lower Bound**: $\mathcal{O}(K^{-\frac{2\alpha\beta}{2\alpha\beta+1}} \cdot T^{\frac{1}{2\alpha\beta+1}} + H^{\frac{2}{2\alpha+1}} \cdot T^{\frac{1}{2\alpha+1}})$



(a) Identical, $d = 10$ (b) Sparse, $d = 10$ (c) Dense, $d = 10$



(d) Identical, $d = 100$ (e) Sparse, $d = 100$ (f) Dense, $d = 100$

Two Transfer Schemes

- **$O2O_{on}$**: each episode recomputes an aggregate estimator from *preceding* episode's source data and debiases using the matching target obs. This persistent adaptation leads to faster regret.
- **$O2O_{off}$**: employs phased transfer only during initial episode, resulting in asymptotic regret growth rates that eventually match single-market learning, though with improved constants.

Online-to-Online

Algorithm 1: CM-TDP-$O2O_{on}$

Input: Streaming source data $\{(p_t^{(k)}, x_t^{(k)}, y_t^{(k)})\}_{t\geq 1}$ for $k \in [K]$; streaming target contexts $\{x_t^{(0)}\}_{t\geq 1}$

Initialisation: $\ell_1 \leftarrow 1$, $\mathcal{T}_1 = \{1\}$, $\widehat{\mathring{g}}_0^{(0)} = 0$.
for $m = 1, 2, \dots$ **do** // episodes
  Compute $\ell_m = 2^{m-1}$, $\mathcal{T}_m = \{\ell_m, \dots, \ell_{m+1} - 1\}$.
  // (i) aggregate previous episode's source data
  $\widehat{\mathring{g}}_m^{(ag)} \leftarrow$ MLE_or_KRR$(\{(p_t^{(k)}, x_t^{(k)}, y_t^{(k)})\}_{t\in\mathcal{T}_{m-1}, k\in[K]})$.
  // (ii) debias with previous episode's target data
  $\widehat{\delta}_m \leftarrow$ Debias$(\widehat{\mathring{g}}_m^{(ag)}, \{(p_t^{(0)}, x_t^{(0)}, y_t^{(0)})\}_{t\in\mathcal{T}_{m-1}})$.
  // For functions MLE_or_KRR and Debias, call Algorithm 2 for linear utility (or Algorithm 3 for non-parametric utility)
  Set $\widehat{\mathring{g}}_m^{(0)} \leftarrow \widehat{\mathring{g}}_m^{(ag)} + \widehat{\delta}_m$.
  for $t \in \mathcal{T}_m$ **do** // episode pricing
    Post price $\widehat{p}_t^{(0)} = h(\widehat{\mathring{g}}_m^{(0)}(x_t^{(0)}))$; observe $y_t^{(0)}$ and store data.

Offline-to-Online

Algorithm 4: CM-TDP-$O2O_{off}$

Input: Offline source market data $\{(p_t^{(k)}, x_t^{(k)}, y_t^{(k)})\}_{t\in\mathcal{H}^{(k)}}$ for $k \in [K]$; feature matrix $\{x_t^{(0)}\}_{t\in\mathbb{N}}$ for the target market

/* ******* Phase 1: Update with transfer learning ******* */
Call Algorithm 2 or 3 to calculate the initial aggregated estimate $\widehat{\mathring{g}}^{(ag)}$ using entire source market data $\{(\boldsymbol{p}_t^{(k)}, \boldsymbol{X}_t^{(k)}, \boldsymbol{y}_t^{(k)})\}_{t\in\mathcal{H}^{(k)}}$ for $k \in [K]$
Apply the price $\widehat{p}_1^{(0)} := h(\widehat{\mathring{g}}^{(ag)}(x_1^{(0)}))$ and collect data $(\widehat{p}_1^{(0)}, x_1^{(0)}, y_1^{(0)})$.
for *each episode* $m = 2, \dots, m_0$ **do**
  Set the length of the m-th episode: $\ell_m := 2^{m-1}$
  Call Algorithm 2 or 3 to calculate the debiasing estimate $\widehat{\delta}_m$ using target market data $\{(p_t^{(0)}, x_t^{(0)}, y_t^{(0)})\}_{t\in[2^{m-2}, 2^{m-1}-1]}$ and aggregated estimate $\widehat{\mathring{g}}^{(ag)}$.
  Set
$$\widehat{\mathring{g}}_m^{(0)} := \widehat{\mathring{g}}^{(ag)} + \widehat{\delta}_m.$$
  For each time t, apply price $\widehat{p}_t^{(0)} := h(\widehat{\mathring{g}}_m^{(0)}(x_t^{(0)}))$ and collect data $(\widehat{p}_t^{(0)}, x_t^{(0)}, y_t^{(0)})$.
/* ******* Phase 2: Update without transfer learning ******* */
for *each* $m \geq m_0 + 1$ **do**
  Set the length of the m-th episode: $\ell_m := 2^{m-1}$
  Call Algorithm 2 or 3 to calculate $\widehat{\mathring{g}}_m^{(0)}$ using target market data $\{(p_t^{(0)}, x_t^{(0)}, y_t^{(0)})\}_{t\in[2^{m-2}, 2^{m-1}-1]}$.
  For each time t, apply price and collect data.

Output: Offered price $\widehat{p}_t^{(0)}$, $t \geq 1$