Electronic Companion for Doubly Adaptive Cascading Bandits with User Abandonment

Junyu Cao
McCombs School of Business, The University of Texas at Austin. junyu.cao@mccombs.utexas.edu
Wei Sun
IBM Research. sunw@us.ibm.com
Zuo-Jun (Max) Shen
Logistics and Supply Chain Management, The University of Hong Kong. maxshen@hku.hk

Appendix EC.1: Additional results

Our proposed algorithm can be extended to sending multiple messages to a single user at a given time. We provide the pseudo-code below. For any active user l with messages scheduled for time t, we compute the optimal subsequence of messages subject to a cardinality constraint (line 5). This subsequence is then sent to the user, after which we collect feedback and update the optimistic estimators accordingly (line 16).

Appendix EC.2: Numerical experiments on real data

EC.2.1. Measure abandonment

We use inactivity from the last interaction with the website as a proxy for abandonment. Data reveals that users interacted frequently with the website, i.e., the 50th and 95th percentile of the time interval between two consecutive visits to the website are 4 hours 24 minutes and 2 days 13 hours and 8 minutes respectively. We believe that such a phenomenon is driven by the fact that 78.8% of users in the dataset are "deep shoppers", i.e., frequent shoppers at Taobao whose purchase frequency and total expenditure have reached the highest status. Thus, for our experiment, we set the threshold on inactivity for abandonment as 3 days. That is, we consider a user "abandoned" the website if it has been more than 3 days of inactivity since her last visit.

To fit the data into our framework, for each user, we construct the entire ad response as a sequence. Depending on when the last interaction takes place, we label whether a user abandoned the website. For instance, a sequence "001100A" indicates 6 interactions a user had with the website, where

Algorithm DAC-Bandit An online learning algorithm for DAC-Bandit with multiple messages

Initialization: Available messages $\mathcal{X}$; set $u_{0i}^{OP} = 1$ for all $i \in \mathcal{X}$ and $q_{0j}^{OP} = 0$ for all $1 \le j \le M$; $t = 1$;
while $t < T$ **do**
  Initialize $\mathcal{O}_t = \emptyset$;
  for Any active user l with messages scheduled to be sent at time t **do**
    Compute $\mathbf{S} = \arg\max_{\mathbf{S}' \subset \mathcal{X} \setminus \mathcal{O}_l} r(\mathbf{S}'; \mathbf{u}_t^{OP}, (q_{t(|\mathcal{O}_l|+1)}^{OP}, \cdots, q_{tM}^{OP}))$ with cardinality constraint $M - |\mathcal{O}_l|$ according to Algorithm OS
    if $|\mathbf{S}| > 0$ and $|\mathcal{O}_l| < M$ **then**
      Send message $\mathbf{S}' = (S_1, \cdots, S_{\min\{\bar{m}, |\mathbf{S}|\}})$ to user l
      $\mathcal{O}_l = \mathcal{O}_l \cup \mathbf{S}'$
    else
      Label user l as inactive
    end if
  end for
  Compute $\mathbf{S}^t = \arg\max_{\mathbf{S}'} r(\mathbf{S}'; \mathbf{u}_t^{OP}, \boldsymbol{q}_t^{OP})$ according to Algorithm OS
  Offer $\mathbf{S}' := (S_1^t, \cdots, S_{\min\{\bar{m}, |\mathbf{S}^t|\}}^t)$ to user t; $\mathcal{O}_t = \mathbf{S}'$
  for Any feedback for some message i as the j^{th} message **do**
    Update u_{ti}^{OP} and q_{tj}^{OP} according to Equation (3) and (4)
  end for
  for Any user l abandons the platform **do**
    Label user l as inactive
  end for
  $t = t + 1$
end while

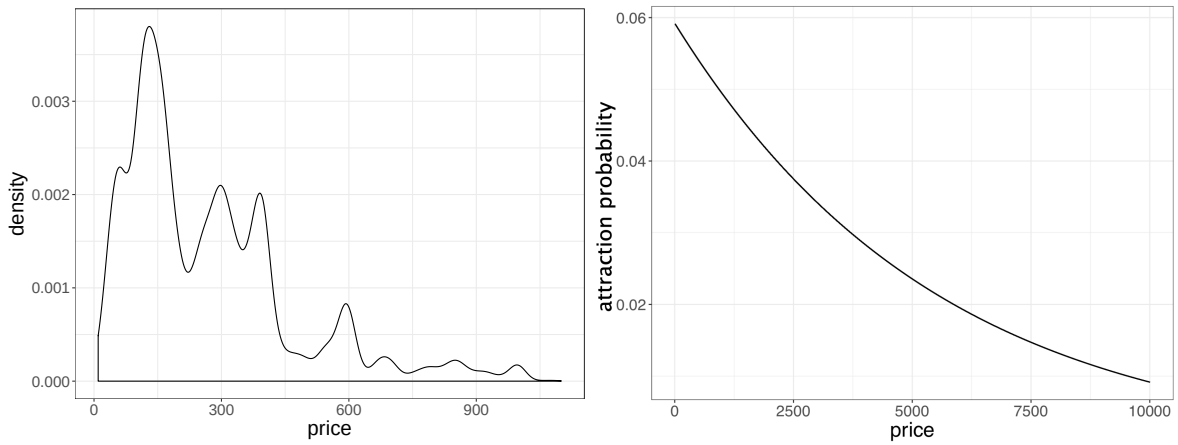
she clicked on the 3rd and 4th ads and abandoned eventually. To handle the situation that a user can select multiple items as the example shown earlier, we separate that response into multiple sequences once a user clicked on an ad, i.e., "001100A" is broken into "001", "1", "00A"[2].

The abandonment rate is defined as the total number of abandonments divided by the total number of ads that are not being clicked. We found that the average abandonment rate is 0.64% in this dataset.

EC.2.2. Measure price effect

Table EC.2 shows the logistic regression result on product valuation. The price distribution and fitted attraction probability are shown in Figure EC.1.

[2] An alternative way to handle such data is to only keep the sequence until a "1" and discard the remaining observations.



(a) (b)

Figure EC.1 Price distribution and the fitted attraction probability.

EC.2.3. Extensive robustness checks across product scope and objective

As a robustness check, we conduct an experiment using the top 100 most popular products across all categories, rather than restricting attention to category 1665. The figure above presents the cumulative regret over time for three algorithms: DAC (red), ε-Greedy (green), and ETC (blue).

Figure EC.2 shows that, DAC continues to achieve the lowest cumulative regret, demonstrating its robustness and effectiveness even when applied to a more diverse and heterogeneous product set. The regret of ε-Greedy grows rapidly and remains substantially higher than that of both DAC and ETC, with a wide confidence band indicating high variability and instability. ETC performs better than ε-Greedy, but still incurs noticeably more regret than DAC throughout the time horizon. The performance gap between DAC and the benchmarks is consistent with the earlier experiment (focused on a single category), suggesting that DAC generalizes well across different product types and sampling scenarios.

In this second robustness experiment, we evaluate the algorithms under a different objective: maximizing total revenue instead of total click-through rate. The plot above compares the cumulative regret over time for DAC (red), ε-Greedy (green), and ETC (blue), with regret now defined relative to the optimal revenue.

Figure EC.3 shows the cumulative regret, where panel (a) reports the regret and panel (b) presents its logarithm for a better visualization. The results indicate that DAC consistently delivers the best performance, maintaining the lowest cumulative regret across all time periods. This indicates that its data acquisition and decision strategy remains effective even when the objective shifts from engagement (clicks) to monetary return (revenue). ε-Greedy suffers from substantial regret,

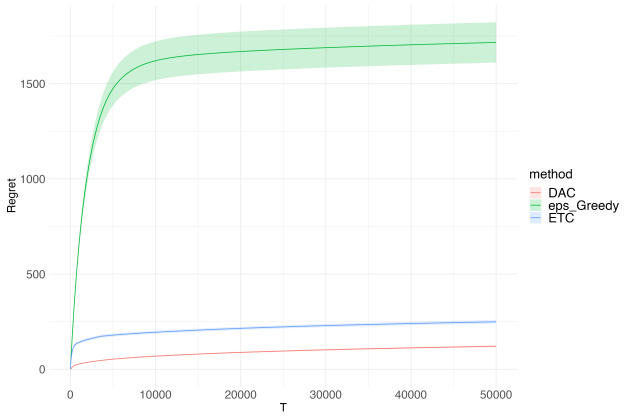
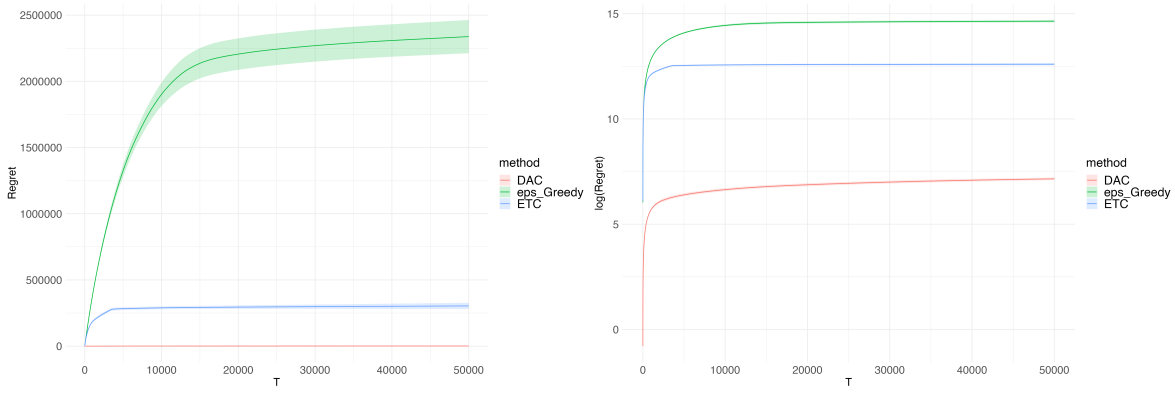


Figure EC.2 Regrets in the multi-category experiment.

again exhibiting fast growth in the early rounds and wide variability. This suggests that its fixed exploration strategy fails to efficiently identify high-revenue options. ETC performs better than ε-Greedy, but still consistently underperforms compared to DAC. Its regret grows more slowly but remains significantly higher. The result confirms that DAC's performance advantage is not tied to a specific reward structure. By incorporating the marginal value of labeling with respect to other utility function (e.g., click or revenue), DAC flexibly adapts its sampling policy to different business objectives.



(a) Regrets of three algorithms. (b) Logarithm of regrets.

Figure EC.3 Regret comparison with revenue objective.

EC.2.4. Sequential update versus batch update

One novelty of our learning problem is that it can dynamically adjust the recommendation for a single user when more feedback is obtained. Meanwhile, in the existing learning-to-rank literature, recommendations are static for an individual user and are updated only across users. We use the

following experiment to compare our proposed sequential update strategy with the batch update method. (The latter updates its strategy only after the entire "batch" or sequence of a user's feedback is received.)

We consider a setting with $N = 50$ and with the cost of abandonment $c = 0.5$. The attraction probability $\mathbf{u}$ is uniformly generated from [0,0.2]. The abandonment distribution is drawn from the truncated Poisson distribution with $q = 10$. In contrast to Algorithm DAC-Bandit, which updates the sequence for all active users whenever feedback is received, a batch update algorithm determines the message sequence for a new user arriving at time t and does not make additional changes to the list.

Figure EC.4 shows the comparison between the two algorithms. The average regret from 50 simulations in the first $T = 50,000$ rounds is 418.13 for the sequential update strategy (Algorithm DAC-Bandit) and 500.30 for the batch update algorithm. Thus, the regret achieved by the sequential update strategy is about 20% lower than that of the batch update approach, demonstrating the benefits of frequent updates on the estimators.

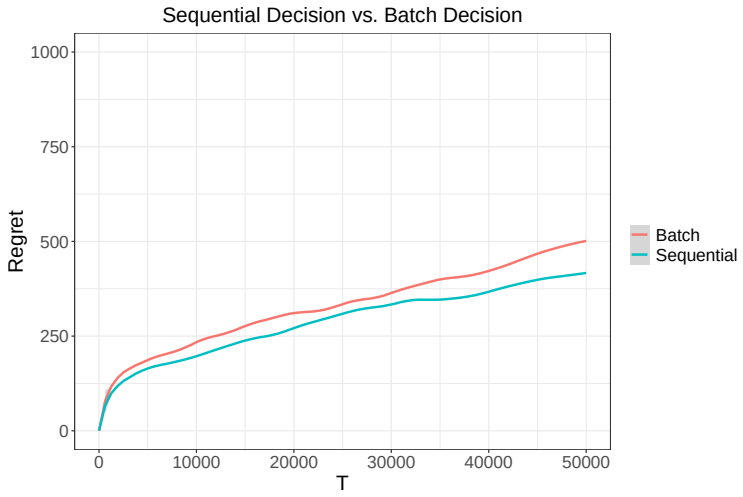


Figure EC.4 Comparison between sequential update and batch update.

Appendix EC.3: Supplementary results

LEMMA EC.1 **(Lemma 3.1 in Tropp (2011))**. *Consider a finite adapted sequence* $\{\mathbf{X}_k\}$ *of positive-semidefinite matrices with dimension* d*, and suppose that*

$$\lambda_{\max}(\mathbf{X}_k) \leq R \text{ almost surely.}$$

Define the finite series

$$\mathbf{Y} := \sum_k \mathbf{X}_k \quad \textit{and} \quad \mathbf{W} := \sum_k \mathbb{E}_{k-1}\mathbf{X}_k.$$

For all $\mu \geq 0$*,*

$$P(\lambda_{\min}(\mathbf{Y}) \leq (1-\delta)\mu \textit{ and } \lambda_{\min}(\mathbf{W}) \geq \mu) \leq d\left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu/R} \textit{ for } \delta \in [0,1), \textit{ and}$$

$$P(\lambda_{\max}(\mathbf{Y}) \geq (1+\delta)\mu \textit{ and } \lambda_{\max}(\mathbf{W}) \leq \mu) \leq d\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu/R} \textit{ for } \delta \geq 0.$$

Table EC.1 Logistic Regression Result on User Abandonment Behavior Parameter $\phi(\mathbf{x})$.

	Coefficient
Gender = Male (ref = Female)	0.122^{***}
	(0.023)
Age_level = 0 (ref = 6)	0.534
	(0.418)
Age_level = 1 (ref = 6)	-0.139
	(0.086)
Age_level = 2 (ref = 6)	-0.263^{***}
	(0.076)
Age_level = 3 (ref = 6)	-0.318^{***}
	(0.075)
Age_level = 4 (ref = 6)	-0.306^{***}
	(0.075)
Age_level = 5 (ref = 6)	-0.234^{***}
	(0.076)
Shopping_level = Shallow (ref = Deep)	0.523^{***}
	(0.159)
Shopping_level= Medium (ref = Deep)	0.303^{***}
	(0.046)
Intercept	-4.827^{***}
	(0.073)

Note: *p<0.1; **p<0.05; ***p<0.01

Table EC.2 Logistic Regression Result on Product Valuation

	Coefficient
Price	$-1.918e{-}04^{***}$
	(1.547e−05)
Intercept	-2.766^{***}
	(0.005)

Note: *p<0.1; **p<0.05; ***p<0.01