EE675 Course Project

RL Algorithms for Personalized Content Recommendation and Real-time Optimization

Instructor : Subramanya Swamy Peruru

Students: Ch Keerthana (210290) , Aeligeti Meghana (210073)

Blog

Video Presentation

Contents

Contextual Bandits paper by yahoo!

Introduction to the paper:

- The paper talks about identifying and presenting the most suitable web-based content to individual users at a given instant of time.
- Given the dynamic nature of content repositories with frequent updates, this task becomes crucial for maintaining user engagement and relevance.
- Traditional methods struggle to scale with the complexity of multivariate optimization, making randomized experiments inefficient.
- The paper introduces a bandit-based approach to efficiently explore web page layout space, considering interactions between components and making real-time content selections.

Contextual Bandits paper by yahoo!

Present day Requirements

- Need for Personalization
- Dynamic Content and User Feedback

Tradtional Approaches:

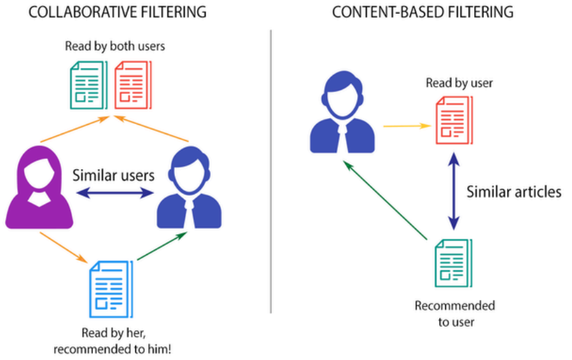
- Epsilon - Greedy
- UCB 1
- EXP 3

How to personalize the content?

- User Features
- Content Features

Traditional Recommender Systems:

- Collaborative filtering
- Content-based filtering
- Hybrid approaches



Some Challenges:

- New-user situation called as ‘cold start’
- Acquiring info can be expensive
- May reduce user satisfaction in the short term

Proposed Approach

- Modelling as Contextual Bandits

Contextual Bandits paper by yahoo!

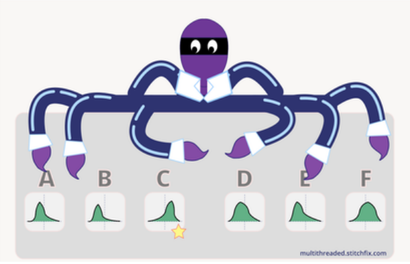
A Multi-armed Bandit Formulation

The Contextual Bandits algorithm proceeds in discrete time trails t = 1, 2, 3, ...

The algorithm at a given trail 't' proceeds as :

1. The algorithm observes the current user u_t and a set $\mathcal{A}_t$ of arms or actions together with their feature vectors $\mathbf{x}_{t,a}$ for $a \in \mathcal{A}_t$. The vector $\mathbf{x}_{t,a}$ summarizes information of *both* the user u_t and arm a, and will be referred to as the *context*.
2. Based on observed payoffs in previous trials, A chooses an arm $a_t \in \mathcal{A}_t$, and receives payoff r_{t,a_t} whose expectation depends on both the user u_t and the arm a_t.
3. The algorithm then improves its arm-selection strategy with the new observation, $(\mathbf{x}_{t,a_t}, a_t, r_{t,a_t})$. It is important to emphasize here that *no* feedback (namely, the payoff $r_{t,a}$) is observed for *unchosen* arms $a \neq a_t$. The consequence of this fact is discussed in more details in the next subsection.

Definitions Used:



$\sum_{t=1}^{T} r_{t,a_t}$ is the *total T-trial payoff*

$\mathbf{E}\left[\sum_{t=1}^{T} r_{t,a_t^*}\right]$ is the *optimal expected T-trial payoff*

a_t^* is the arm with maximum expected payoff at trial t

T-trial regret $R_{\mathsf{A}}(T)$

$$R_{\mathsf{A}}(T) \stackrel{\text{def}}{=} \mathbf{E}\left[\sum_{t=1}^{T} r_{t,a_t^*}\right] - \mathbf{E}\left[\sum_{t=1}^{T} r_{t,a_t}\right]$$

Special case of contextual bandits is the usual K-armed Multi Arm Bandits which is the context-free case

Contextual Bandits paper by yahoo!

We define CTR as Click-Through Rate. Choosing an article with maximum CTR is equivalent to maximizing the expected number of clicks from users, which in turn is the same as maximizing the total expected payoff in our bandit formulation.

Existing Bandit Algorithms

The fundamental challenge is to balance exploration and exploration

- Epsilon - Greedy and it's variants
- UCBs - Upper Confidence Bound
- EXP-4
- Epoch greedy
- LinRel
- Gittens Index method - based on Baye's rule

Proposed Algorithm : LinUCB

We show that a confidence interval can be computed efficiently in closed form when the payoff model is linear, and call this algorithm LinUCB

This has two variants

- LinUCB with Disjoint Linear Models
- LinUCB with Hybrid Linear Models

A. LinUCB with Disjoint Linear Models

We assume the expected payoff of an arm a is linear in its d-dimensional feature $\mathbf{x}_{t,a}$ with some unknown coefficient vector $\boldsymbol{\theta}_a^*$ namely, for all t,

$$\mathbf{E}[r_{t,a}|\mathbf{x}_{t,a}] = \mathbf{x}_{t,a}^\top \boldsymbol{\theta}_a^*$$

Contextual Bandits paper by yahoo!

Let $\mathbf{D}_a$ be a design matrix of dimension m × d at trial t, whose rows correspond to m training inputs (e.g., m contexts that are observed previously for article a), and the corresponding response vector $\mathbf{b}_a \in \mathbb{R}^m$ (e.g., the corresponding m click/no-click user feedback). Applying ridge regression to the training data $(\mathbf{D}_a, \mathbf{c}_a)$ gives an estimate of the coefficients as follows:

$$\hat{\boldsymbol{\theta}}_a = (\mathbf{D}_a^\top \mathbf{D}_a + \mathbf{I}_d)^{-1} \mathbf{D}_a^\top \mathbf{c}_a$$

where $\mathbf{I}_d$ is the $d \times d$ identity matrix. When components in $\mathbf{c}_a$ are independent conditioned on corresponding rows in $\mathbf{D}_a$, it can be shown [27] that, with probability at least $1 - \delta$,

$$\left| \mathbf{x}_{t,a}^\top \hat{\boldsymbol{\theta}}_a - \mathbf{E}[r_{t,a} | \mathbf{x}_{t,a}] \right| \leq \alpha \sqrt{\mathbf{x}_{t,a}^\top (\mathbf{D}_a^\top \mathbf{D}_a + \mathbf{I}_d)^{-1} \mathbf{x}_{t,a}}$$

for any $\delta > 0$ and $\mathbf{x}_{t,a} \in \mathbb{R}^d$, where $\alpha = 1 + \sqrt{\ln(2/\delta)/2}$ is a constant. In other words, the inequality above gives a reasonably tight UCB for the expected payoff of arm a, from which a UCB-type arm-selection strategy can be derived: at each trial t, choose

$$a_t \stackrel{\text{def}}{=} \arg\max_{a \in \mathcal{A}_t} \left(\mathbf{x}_{t,a}^\top \hat{\boldsymbol{\theta}}_a + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}} \right)$$

$$\text{where } \mathbf{A}_a \stackrel{\text{def}}{=} \mathbf{D}_a^\top \mathbf{D}_a + \mathbf{I}_d$$

The confidence interval may be motivated and derived from other principles also. For instance, ridge regression can also be interpreted as a Bayesian point estimate.

Here in our model, the only input parameter is α

Contextual Bandits paper by yahoo!

Algorithm 1 LinUCB with disjoint linear models.

Inputs: $\alpha \in \mathbb{R}_+$
for $t = 1, 2, 3, \ldots, T$ **do**
  Observe features of all arms $a \in \mathcal{A}_t$: $\mathbf{x}_{t,a} \in \mathbb{R}^d$
  for all $a \in \mathcal{A}_t$ **do**
    if a is new **then**
      $\mathbf{A}_a \leftarrow \mathbf{I}_d$ (d-dimensional identity matrix)
      $\mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1}$ (d-dimensional zero vector)
    end if
    $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1}\mathbf{b}_a$
    $p_{t,a} \leftarrow \hat{\boldsymbol{\theta}}_a^\top \mathbf{x}_{t,a} + \alpha\sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1}\mathbf{x}_{t,a}}$
  end for
  Choose arm $a_t = \arg\max_{a\in\mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
  $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^\top$
  $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t\mathbf{x}_{t,a_t}$
end for

B. LinUCB with Hybrid Linear Models

$$\mathbf{E}[r_{t,a}|\mathbf{x}_{t,a}] = \mathbf{z}_{t,a}^\top\boldsymbol{\beta}^* + \mathbf{x}_{t,a}^\top\boldsymbol{\theta}_a^*$$

where $\mathbf{z}_{t,a} \in \mathbb{R}^k$ is the feature of the current user/article combination, and $\boldsymbol{\beta}^*$ is an unknown coefficient vector common to all arms. This model is hybrid in the sense that some of the coefficients $\boldsymbol{\beta}^*$ are shared by all arms, while others $\boldsymbol{\theta}_a^*$ are not.

For hybrid models, we can no longer use Algorithm 1 as the confidence intervals of various arms are not independent due to the shared features.

There is an efficient way to compute an UCB along the same line of reasoning and derivation relies heavily on block matrix inversion techniques.

Contextual Bandits paper by yahoo!

Algorithm 2 LinUCB with hybrid linear models.

Inputs: $\alpha \in \mathbb{R}_+$
$\mathbf{A}_0 \leftarrow \mathbf{I}_k$ (k-dimensional identity matrix)
$\mathbf{b}_0 \leftarrow \mathbf{0}_k$ (k-dimensional zero vector)
for $t = 1, 2, 3, \ldots, T$ **do**
  Observe features of all arms $a \in \mathcal{A}_t$: $(\mathbf{z}_{t,a}, \mathbf{x}_{t,a}) \in \mathbb{R}^{k+d}$
  $\hat{\boldsymbol{\beta}} \leftarrow \mathbf{A}_0^{-1}\mathbf{b}_0$
  for all $a \in \mathcal{A}_t$ **do**
    if a is new **then**
      $\mathbf{A}_a \leftarrow \mathbf{I}_d$ (d-dimensional identity matrix)
      $\mathbf{B}_a \leftarrow \mathbf{0}_{d\times k}$ (d-by-k zero matrix)
      $\mathbf{b}_a \leftarrow \mathbf{0}_{d\times 1}$ (d-dimensional zero vector)
    end if
    $\hat{\boldsymbol{\theta}}_a \leftarrow \mathbf{A}_a^{-1}\left(\mathbf{b}_a - \mathbf{B}_a\hat{\boldsymbol{\beta}}\right)$
    $s_{t,a} \leftarrow \mathbf{z}_{t,a}^\top \mathbf{A}_0^{-1}\mathbf{z}_{t,a} - 2\mathbf{z}_{t,a}^\top \mathbf{A}_0^{-1}\mathbf{B}_a^\top \mathbf{A}_a^{-1}\mathbf{x}_{t,a} + \mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1}\mathbf{x}_{t,a} + \mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1}\mathbf{B}_a\mathbf{A}_0^{-1}\mathbf{B}_a^\top \mathbf{A}_a^{-1}\mathbf{x}_{t,a}$
    $p_{t,a} \leftarrow \mathbf{z}_{t,a}^\top\hat{\boldsymbol{\beta}} + \mathbf{x}_{t,a}^\top\hat{\boldsymbol{\theta}}_a + \alpha\sqrt{s_{t,a}}$
  end for
  Choose arm $a_t = \arg\max_{a\in\mathcal{A}_t} p_{t,a}$ with ties broken arbitrarily, and observe a real-valued payoff r_t
  $\mathbf{A}_0 \leftarrow \mathbf{A}_0 + \mathbf{B}_{a_t}^\top \mathbf{A}_{a_t}^{-1}\mathbf{B}_{a_t}$
  $\mathbf{b}_0 \leftarrow \mathbf{b}_0 + \mathbf{B}_{a_t}^\top \mathbf{A}_{a_t}^{-1}\mathbf{b}_{a_t}$
  $\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{x}_{t,a_t}^\top$
  $\mathbf{B}_{a_t} \leftarrow \mathbf{B}_{a_t} + \mathbf{x}_{t,a_t}\mathbf{z}_{t,a_t}^\top$
  $\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t\mathbf{x}_{t,a_t}$
  $\mathbf{A}_0 \leftarrow \mathbf{A}_0 + \mathbf{z}_{t,a_t}\mathbf{z}_{t,a_t}^\top - \mathbf{B}_{a_t}^\top \mathbf{A}_{a_t}^{-1}\mathbf{B}_{a_t}$
  $\mathbf{b}_0 \leftarrow \mathbf{b}_0 + r_t\mathbf{z}_{t,a_t} - \mathbf{B}_{a_t}^\top \mathbf{A}_{a_t}^{-1}\mathbf{b}_{a_t}$
end for

- Like all UCB methods, LinUCB always chooses the arm with highest UCB
- Computational complexity is linear in the number of arms and at most cubic in the number of features
- To decrease computation further, we may update $\mathbf{A}_{a_t}$ in every step, which takes $O(d^2)$ time, but compute and cache $\mathbf{Q}_a \overset{\text{def}}{=} \mathbf{A}_a^{-1}$ (for all a) periodically instead of in real time
- The algorithm works well for a dynamic arm set, and remains efficient as long as the size of $\mathbf{A}_{a_t}$ is not too large
- The regret bound is of order $\tilde{O}(\sqrt{KdT})$
- Final computational complexity after reducing per-trial computation by using cache updates periodically rather than at the end of each trial is of the order $O(d^2 + k^2)$

Contextual Bandits paper by yahoo!

Evaluation Methodology

- It is difficult to evaluate a bandit policy
- Can you we run on live data?
- But, we won't due to logistical challenges
- Use offline data that was collected previosuly
- But it was with entirely different policy
- Hence, we use the concept of "off-policy" evaluation

We may build a simulator to model the bandit process from the logged data, and then evaluate π with the simulator.

But, the modeling step will introduce bias in the simulator and so make it hard to justify the reliability of this simulator-based evaluation

Proposed Evaluation Algorithm:

In this section, we describe a provably reliable technique for carrying out such an evaluation, assuming that the individual events are i.i.d., and that the logging policy that was used to gather the logged data chose each arm at each time step uniformly at random.

More precisely, we suppose that there is some unknown distribution D from which tuples are drawn i.i.d. of the form $(\mathbf{x}_1, ..., \mathbf{x}_K, r_1, \ldots, r_K)$, each consisting of observed feature vectors and *hidden* payoffs for all arms. We also posit access to a large sequence of logged events resulting from the interaction of the logging policy with the world. Each such event consists of the context vectors $\mathbf{x}_1, ..., \mathbf{x}_K$, a selected arm a and the resulting observed payoff r_a. Crucially, only the payoff r_a is observed for the single arm a that was chosen uniformly at random. For simplicity of presentation, we take this sequence of logged events to be an infinitely long stream; however, we also give explicit bounds on the actual finite number of events required by our evaluation method.

Our goal is to use this data to evaluate a bandit algorithm π. Formally, π is a (possibly randomized) mapping for selecting the arm a_t at time t based on the history h_{t-1} of $t-1$ preceding events, together with the current context vectors $\mathbf{x}_{t1}, ..., \mathbf{x}_{tK}$.

Contextual Bandits paper by yahoo!

Our proposed policy evaluator is shown in Algorithm 3. The method takes as input a policy π and a desired number of "good" events T on which to base the evaluation. We then step through the stream of logged events one by one. If, given the current history h_{t-1}, it happens that the policy π chooses the same arm a as the one that was selected by the logging policy, then the event is retained, that is, added to the history, and the total payoff R_t updated. Otherwise, if the policy π selects a different arm from the one that was taken by the logging policy, then the event is entirely ignored, and the algorithm proceeds to the next event without any other change in its state.

Note that, because the logging policy chooses each arm uniformly at random, each event is retained by this algorithm with probability exactly $1/K$, independent of everything else. This means that the events which are retained have the same distribution as if they were selected by D. As a result, we can prove that two processes are equivalent: the first is evaluating the policy against T real-world events from D, and the second is evaluating the policy using the policy evaluator on a stream of logged events.

THEOREM 1. *For all distributions D of contexts, all policies π, all T, and all sequences of events h_T,*

$$\Pr_{\textit{Policy\_Evaluator}(\pi,S)}(h_T) = \Pr_{\pi,D}(h_T)$$

where S is a stream of events drawn i.i.d. from a uniform random logging policy and D. Furthermore, the expected number of events obtained from the stream to gather a history h_T of length T is KT.

This theorem says that *every* history h_T has the identical probability in the real world as in the policy evaluator. Many statistics of these histories, such as the average payoff R_T/T returned by Algorithm 3, are therefore unbiased estimates of the value of the algorithm π. Further, the theorem states that KT logged events are required, in expectation, to retain a sample of size T.

The proof of this theorem is by induction.

Contextual Bandits paper by yahoo!

PROOF. The proof is by induction on $t = 1, \ldots, T$ starting with a base case of the empty history which has probability 1 when $t = 0$ under both methods of evaluation. In the inductive case, assume that we have for all $t - 1$:

$$\Pr_{\text{Policy\_Evaluator}(\pi,S)}(h_{t-1}) = \Pr_{\pi,D}(h_{t-1})$$

and want to prove the same statement for any history h_t. Since the data is i.i.d. and any randomization in the policy is independent of randomization in the world, we need only prove that conditioned on the history h_{t-1} the distribution over the t-th event is the same for each process. In other words, we must show:

$$\Pr_{\text{Policy\_Evaluator}(\pi,S)}((\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,K}, a, r_{t,a}) \mid h_{t-1})$$
$$= \Pr_{D}(\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,K}, r_{t,a}) \Pr_{\pi(h_{t-1})}(a \mid \mathbf{x}_{t,1}, ..., \mathbf{x}_{t,K}).$$

Since the arm a is chosen uniformly at random in the logging policy, the probability that the policy evaluator exits the inner loop is identical for any policy, any history, any features, and any arm, implying this happens for the last event with the probability of the last event, $\Pr_D(\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,K}, r_{t,a})$. Similarly, since the policy π's distribution over arms is independent conditioned on the history h_{t-1} and features $(\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,K})$, the probability of arm a is just $\Pr_{\pi(h_{t-1})}(a|\mathbf{x}_{t,1}, ..., \mathbf{x}_{t,K})$.

Finally, since each event from the stream is retained with probability exactly $1/K$, the expected number required to retain T events is exactly KT. □

Algorithm 3 Policy_Evaluator.

Inputs: $T > 0$; policy π; stream of events
$h_0 \leftarrow \emptyset$ {An initially empty history}
$R_0 \leftarrow 0$ {An initially zero total payoff}
for $t = 1, 2, 3, \ldots, T$ **do**
  repeat
    Get next event $(\mathbf{x}_1, ..., \mathbf{x}_K, a, r_a)$
  until $\pi(h_{t-1}, (\mathbf{x}_1, ..., \mathbf{x}_K)) = a$
  $h_t \leftarrow \text{CONCATENATE}(h_{t-1}, (\mathbf{x}_1, ..., \mathbf{x}_K, a, r_a))$
  $R_t \leftarrow R_{t-1} + r_a$
end for
Output: R_T/T

Contextual Bandits paper by yahoo!

Yahoo! Today Module

A snapshot of the “Featured” tab in the Today Module on Yahoo! Front Page. By default, the article at F1 position is highlighted at the story position.

Experimental Setup:

A. Data Collection

- Data was collected from a randomly selected user group, where articles were also randomly chosen from the article pool for display, focusing exclusively on interactions at the F1 story position to avoid exposure bias.
- Approximately 4.7 million user interaction events from a single day served as "tuning data" to optimize the parameters for competing bandit algorithms.
- These tuned algorithms were subsequently evaluated using a "evaluation data" set containing around 36 million events over a week-long period, assessing their effectiveness.

Contextual Bandits paper by yahoo!

Experimental Setup:

B. Feature Construction

- Feature Construction and Selection: High support raw feature vectors over 1000 categorical components for users and approximately 100 for articles are used, focusing on demographic, geographic, URL, and editorial data.
- Normalization and Encoding: Features are selected, encoded as binary vectors, normalized to unit length, and augmented with a constant feature.
- Dimensionality Reduction and Clustering: Logistic regression models click probabilities; user features are projected into a new space and clustered into five groups using K-means based on article preferences.

- Feature Vector Development for LinUCB: Six-dimensional feature vectors from reduced user and article features are constructed to support testing of disjoint and hybrid LinUCB models, optimizing interaction analysis.

- We first used logistic regression (LR) to fit a bilinear model for click probability given raw user/article features so that $\boldsymbol{\phi}_u^\top \mathbf{W} \boldsymbol{\phi}_a$ approximated the probability that the user u clicks on article a, where $\boldsymbol{\phi}_u$ and $\boldsymbol{\phi}_a$ were the corresponding feature vectors, and $\mathbf{W}$ was a weight matrix optimized by LR.
- Raw user features were then projected onto an induced space by computing $\boldsymbol{\psi}_u \stackrel{\text{def}}{=} \boldsymbol{\phi}_u^\top \mathbf{W}$. Here, the i^{th} component in $\boldsymbol{\psi}_u$ for user u may be interpreted as the degree to which the user likes the i^{th} category of articles. K-means was applied to group users in the induced $\boldsymbol{\psi}_u$ space into 5 clusters.
- The final user feature was a six-vector: five entries corresponded to membership of that user in these 5 clusters (computed with a Gaussian kernel and then normalized so that they sum up to unity), and the sixth was a constant feature 1.

Contextual Bandits paper by yahoo!

Compared Algorithms:

The algorithms empirically evaluated in our experiments can be categorized into three groups:

1. Algorithms that make no use of features
    - Random
    - Ɛ-Greedy
    - UCB
    - Omniscient: Always selects the article with the highest historical CTR.

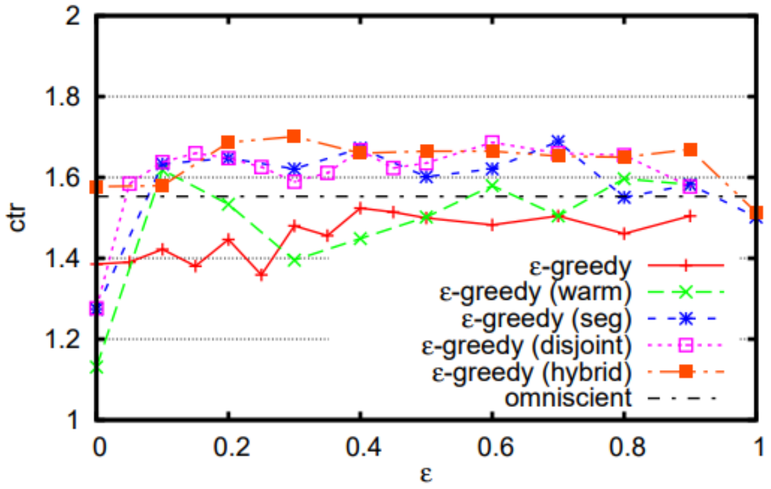
2. Algorithms with “warm start” - an intermediate step towards personlized service
    - Ɛ-Greedy (warm)
    - UCB (warm)

3. Algorithms that learn user-specific CTRs online
    - Ɛ-Greedy (seg)
    - UCB (seg)
    - Ɛ-Greedy (disjoint)
    - LinUCB - our Algorithm
    - Ɛ-Greedy (hybrid)

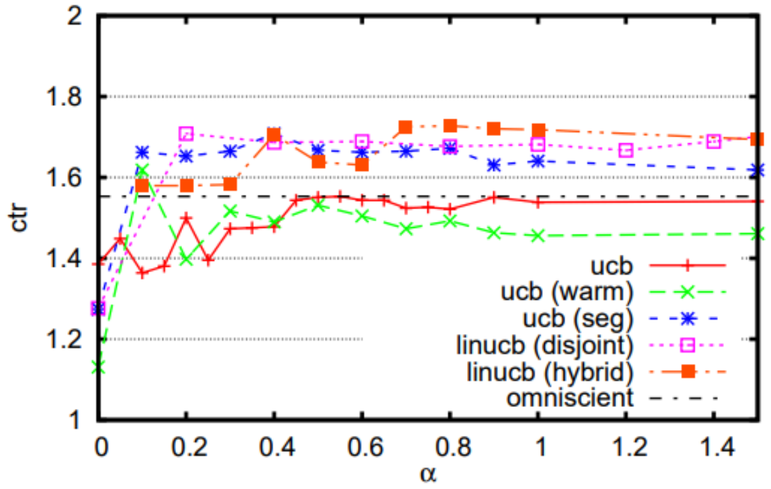
Experimentation and Results:

- Learning Bucket: Smaller portion of traffic used for algorithms to learn and estimate article Click-Through Rates (CTRs)
- Deployment Bucket: Larger portion where articles are selected based on learned CTR estimates to maximize user clicks
- Performance in both buckets is crucial.
- CTRs from both buckets are analyzed to assess the effectiveness of the algorithms.

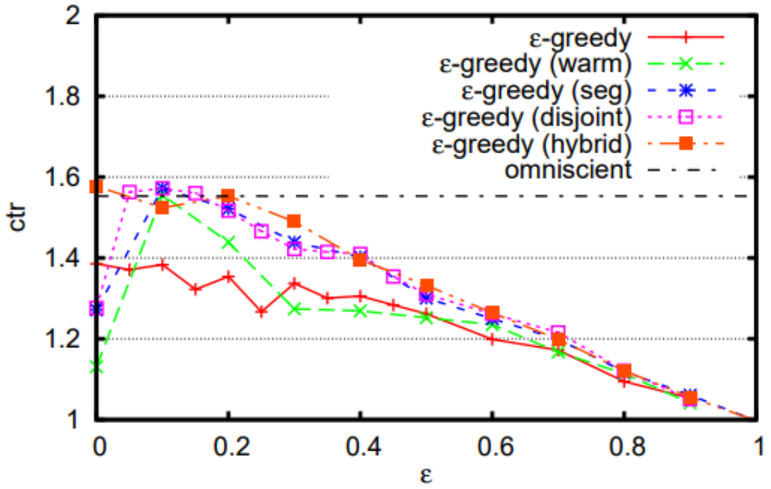
Contextual Bandits paper by yahoo!



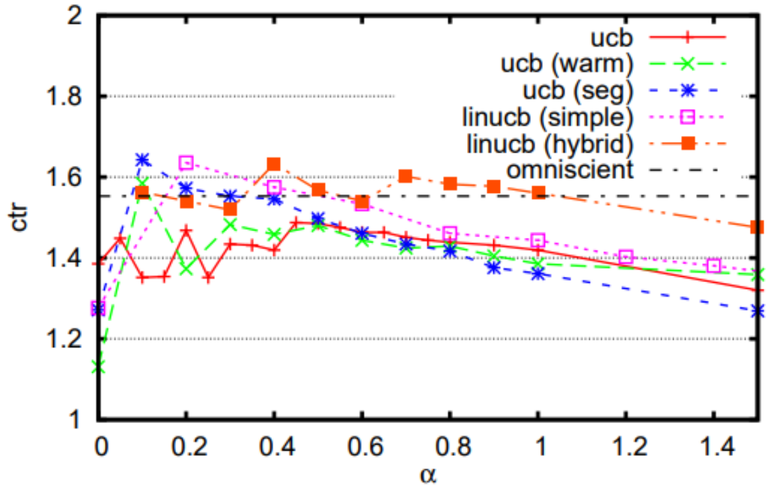
(a) Deployment bucket.



(b) Deployment bucket.



(c) Learning bucket.



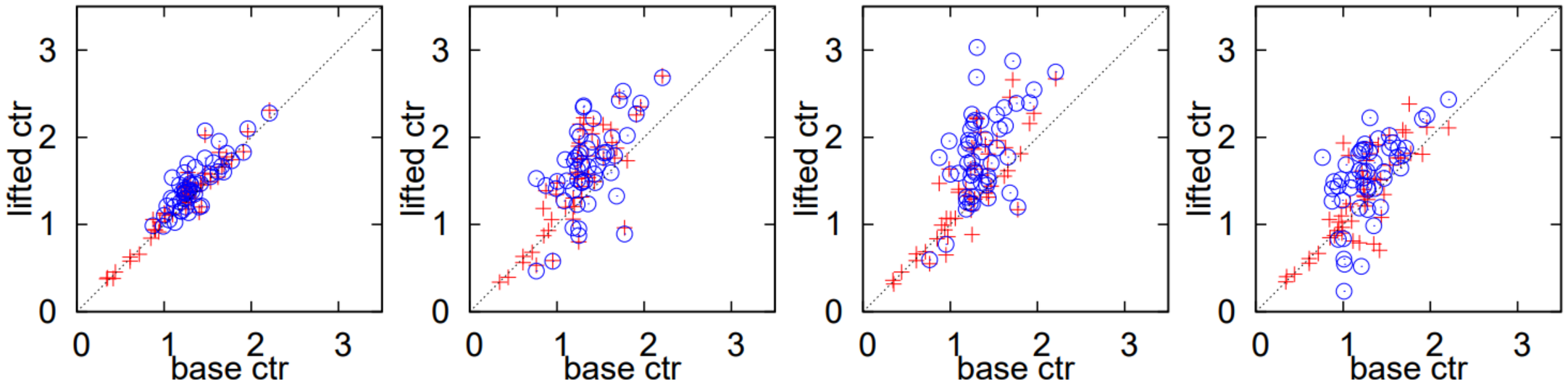
(d) Learning bucket.

Contextual Bandits paper by yahoo!

algorithm	size = 100%		size = 30%		size = 20%		size = 10%		size = 5%		size = 1%	
	deploy	learn	deploy	learn	deploy	learn	deploy	learn	deploy	learn	deploy	learn
ϵ-greedy	1.596 0%	1.326 0%	1.541 0%	1.326 0%	1.549 0%	1.273 0%	1.465 0%	1.326 0%	1.409 0%	1.292 0%	1.234 0%	1.139 0%
ucb	1.594 0%	1.569 18.3%	1.582 2.7%	1.535 15.8%	1.569 1.3%	1.488 16.9%	1.541 5.2%	1.446 9%	1.541 9.4%	1.465 13.4%	1.354 9.7%	1.22 7.1%
ϵ-greedy (seg)	1.742 9.1%	1.446 9%	1.652 7.2%	1.46 10.1%	1.585 2.3%	1.119 −12%	1.474 0.6%	1.284 −3.1%	1.407 0%	1.281 −0.8%	1.245 0.9%	1.072 −5.8%
ucb (seg)	1.781 11.6%	1.677 26.5%	1.742 13%	1.555 17.3%	1.689 9%	1.446 13.6%	1.636 11.7%	1.529 15.3%	1.532 8.7%	1.32 2.2%	1.398 13.3%	1.25 9.7%
ϵ-greedy (disjoint)	1.769 10.8%	1.309 −1.2%	1.686 9.4%	1.337 0.8%	1.624 4.8%	1.529 20.1%	1.529 4.4%	1.451 9.4%	1.432 1.6%	1.345 4.1%	1.262 2.3%	1.183 3.9%
linucb (disjoint)	1.795 12.5%	1.647 24.2%	1.719 11.6%	1.507 13.7%	1.714 10.7%	1.384 8.7%	1.655 13%	1.387 4.6%	1.574 11.7%	1.245 −3.5%	1.382 12%	1.197 5.1%
ϵ-greedy (hybrid)	1.739 9%	1.521 14.7%	1.68 9%	1.345 1.4%	1.636 5.6%	1.449 13.8%	1.58 7.8%	1.348 1.7%	1.465 4%	1.415 9.5%	1.342 8.8%	1.2 5.4%
linucb (hybrid)	1.73 8.4%	1.663 25.4%	1.691 9.7%	1.591 20%	1.708 10.3%	1.619 27.2%	1.675 14.3%	1.535 15.8%	1.588 12.7%	1.507 16.6%	1.482 20.1%	1.446 27%

Table 1: Performance evaluation: CTRs of all algorithms on the one-week evaluation dataset in the deployment and learning buckets (denoted by "deploy" and "learn" in the table, respectively). The numbers with a percentage is the CTR lift compared to ϵ-greedy.

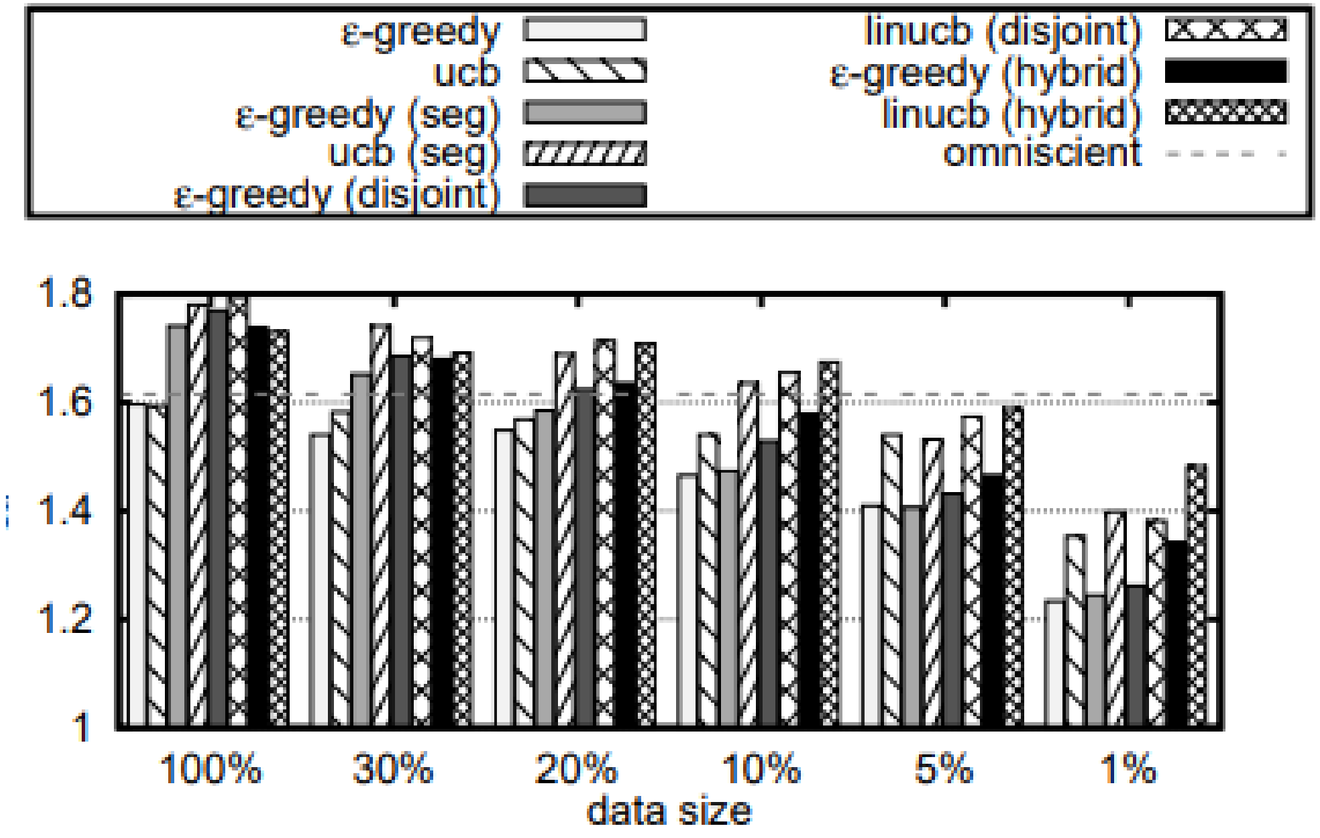
Contextual Bandits paper by yahoo!



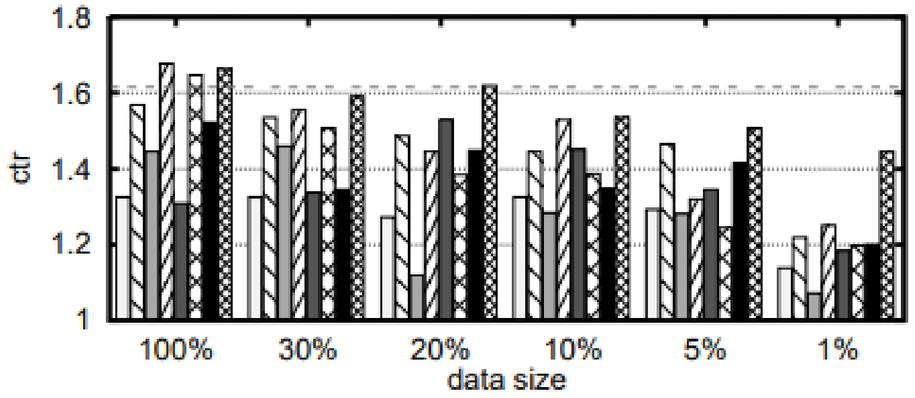
(a) ϵ-greedy and ucb (b) seg:ϵ-greedy and ucb (c) disjoint:ϵ-greedy and linucb (d) hybrid:ϵ-greedy and linucb

Figure 3: Scatterplots of the base CTR vs. lifted CTR (in the learning bucket) of the 50 most frequently selected articles when 100% evaluation data were used. Red crosses are for ϵ-greedy algorithms, and blue circles are for UCB algorithms. Note that the sets of most frequently chosen articles varied with algorithms; see the text for details.

Contextual Bandits paper by yahoo!



(a) CTRs in the deployment bucket.



(b) CTRs in the learning bucket.

Figure 4: CTRs in evaluation data with varying data sizes.

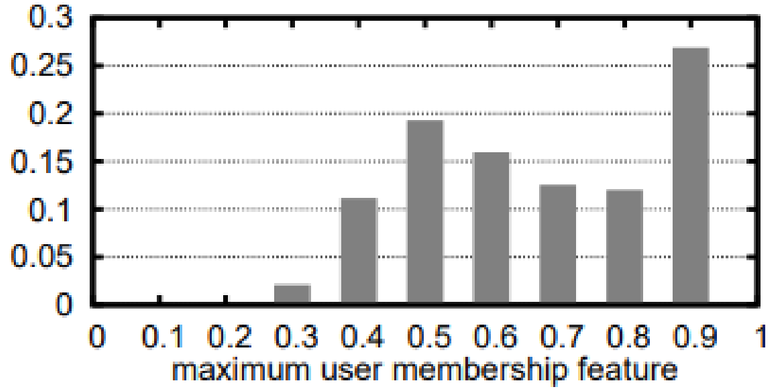


Figure 5: User maximum membership histogram.

An Efficient Bandit Algorithm for Realtime Multivariate Optimization by amazon

Abstract

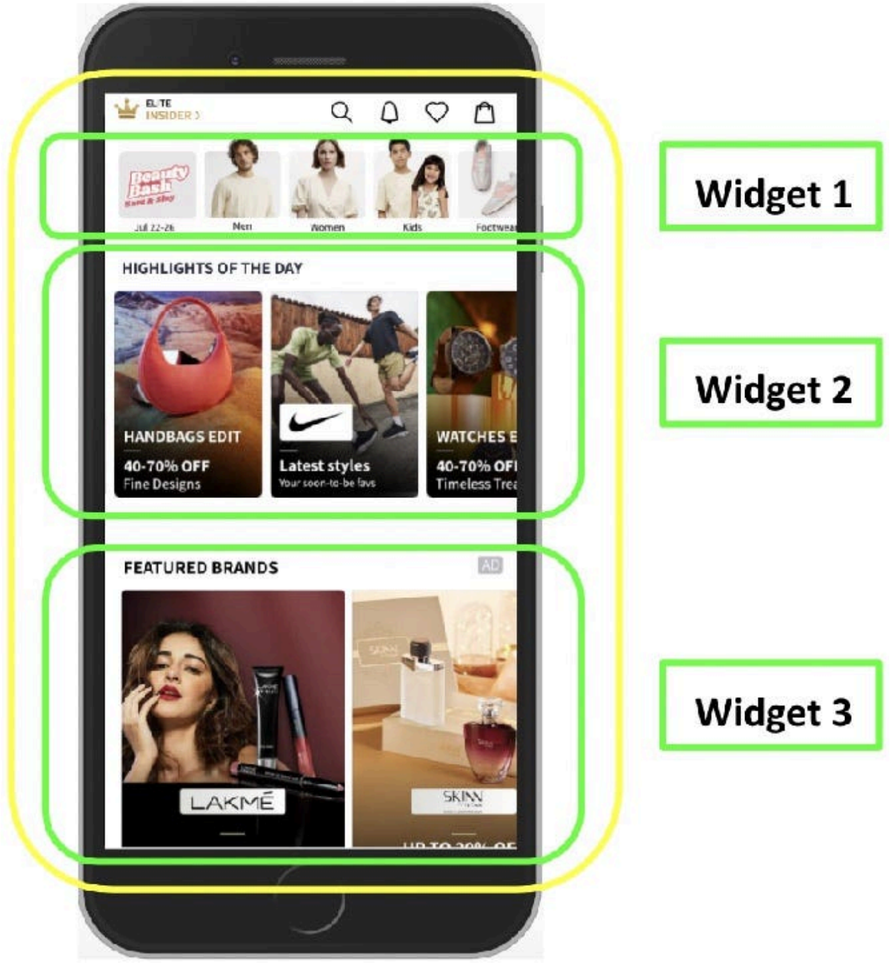
- Web page optimization involves navigating a complex decision space with various elements like layout, images, and text.
- Traditional methods struggle to scale with the complexity of multivariate optimization, making randomized experiments inefficient.
- The paper introduces a bandit-based approach to efficiently explore web page layout space, considering interactions between components and making real-time content selections.
- Demonstrated improvements in conversion rates through simulations and real-world applications highlight the significant impact of the proposed approach, such as a 21% conversion increase in an Amazon service promotion within a week

Introduction & Background

- Complex Web Page Decisions
- Optimization Challenges
- Proposed Approach
- Deployment and Results

Figure 1: Example of a generic promotional message for an Amazon service. Each component is a separate widget with the indicated number of alternative content. There are 48 total distinct layouts.

Layout 1



Problem Setting

Variables

- Layout - A
- Context - X
- Reward - R
- Number of widgets - D
- Number of variations each widget has - N
- A[i] - The content choosen for ith widget
- Feature vector - $B_{A,X}$
- $R_{A,X}$ - Reward for given layout and context

The reward is modelled with a generalized linear model

$$\mathbb{E}[R|A, X] = g(B_{A,X}^{\top}\mu),$$

Optimal arm at time t

$$A_t^* = \underset{A}{argmax}\ E[R_{A,X_t}].$$

Difference between the expected reward of the optimal arm and the selected arm at time t

$$\Delta_t = E[R_{A_t^*,X_t}] - E[R_{A_t,X_t}].$$

Cumulative regret over T rounds

$$\Delta^T = \sum_{t=1}^{T} \Delta_t.$$

Probability Model

We choose the probit function as our link function

$$P(R|A,X) = \Phi\left(R * B_{A,X}^{\top} W\right)$$

Ignoring context X, our linear model becomes:(MVT2)

$$B_A^{\top} W = W^0 + \sum_{i=1}^{D} W_i^1(A) + \sum_{j=1}^{D} \sum_{k=j+1}^{D} W_{j,k}^2(A)$$

Weight class	Definition
W^0	Bias weight
$W_i^1(A)$	Impact of content in i^{th} widget
$W_{i,j}^2(A)$	Interaction of content in i^{th} widget with content in j^{th} widget
$W_i^c(X)$	Impact of i^{th} contextual feature
$W_{i,j}^{1c}(A,X)$	Interaction of content in i^{th} widget with the j^{th} contextual feature
$W^L(A)$	Weight associated with distinct layout A

To account for contextual information X and possible interactions between web page content and context, additional terms can be added to $B_{A,K}$ (MVT2c):

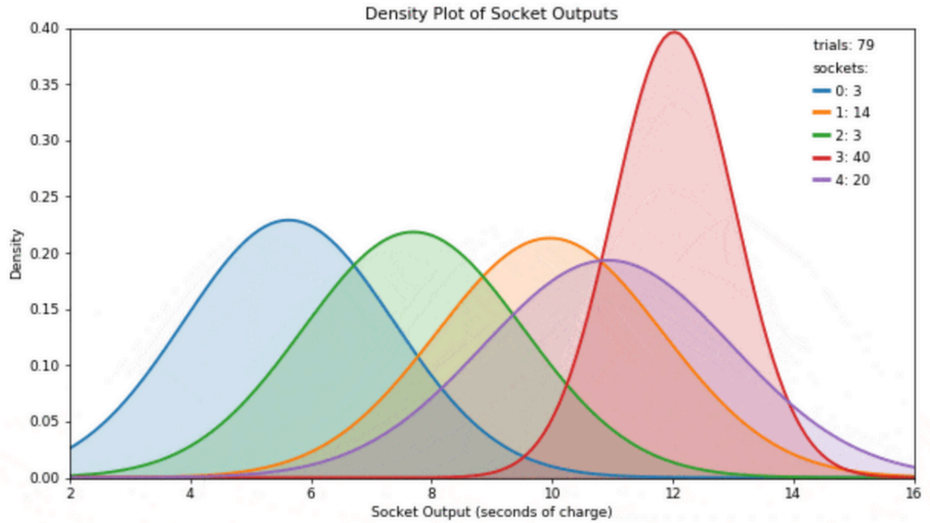
$$B_{A,X}^{\top} W = W^0 + \sum_{i=1}^{D} W_i^1(A) + \sum_{j=1}^{D} \sum_{k=j+1}^{D} W_{j,k}^2(A) + \sum_{l=1}^{L} W_l^c(X) + \sum_{m=1}^{D} \sum_{n=1}^{L} W_{m,n}^{1c}(A,X)$$

Multivariate Algorithm

Thompson Sampling

Algorithm 1 Thompson Sampling for Contextual Bandits

1: **for all** $t = 1, \ldots, T$ **do**
2: Receive context X_t
3: Sample $\tilde{W}_t$ from the posterior $P(W|\mathcal{H}_{t-1})$
4: Select $A_t = argmax_A \; B^{\top}_{A,X_t} \tilde{W}_t$
5: Display layout A_t and observe reward R_t
6: Update $\mathcal{H}_t = \mathcal{H}_{t-1} \cup (A_t, R_t, X_t)$



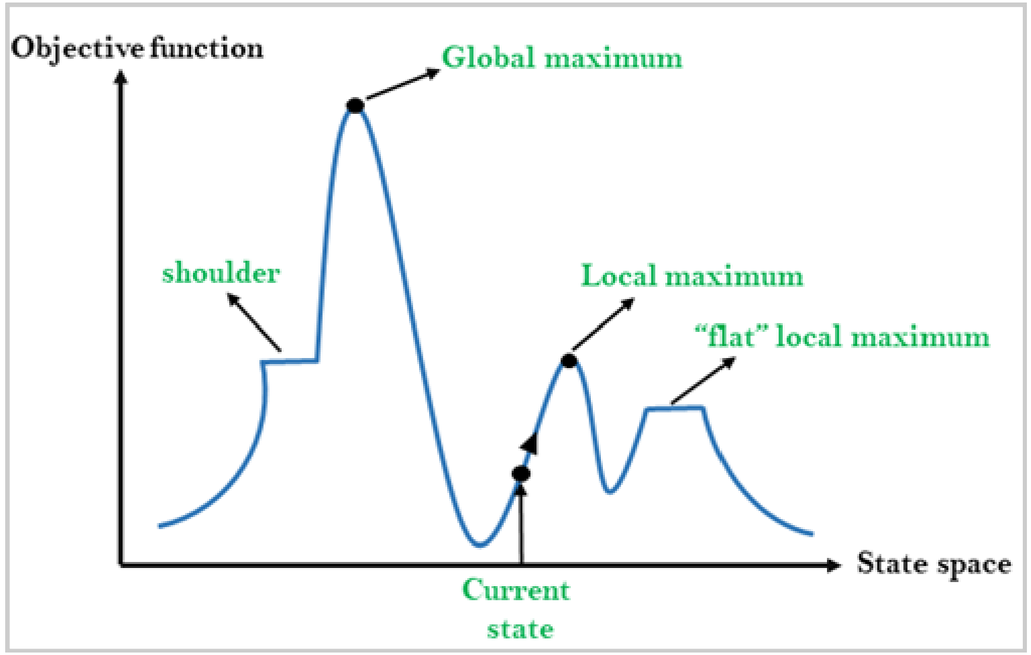
O(ND)

Multivariate Algorithm

Hill Climbing Optimization

Algorithm 2 Hill climbing with random restarts

function HILL CLIMBING SEARCH($\tilde{W}, X$)
  for $s = 1, \ldots, S$ **do**
    Pick a layout A_s^0 randomly
    for $k = 1, \ldots, K$ **do**
      Randomly choose a widget i to optimize
      Find $j^* = argmax_j \, B^\top_{A_s^{k-1} \leftarrow (A[i]=j), X} \tilde{W}$
      $A_s^k = A_s^{k-1} \leftarrow (A[i] = j^*)$
  $s^* = argmax_s \, B^\top_{A_s^K} \tilde{W}$
  return $A_{s^*}^K$



Simulated Data

MVT 1:

$$B_A^T W = W^0 + \sum_{i=1}^{D} W_i^1(A)$$

N^D-MAB :

$$B_A^T W = W^L(A)$$

D-MABs

$$B_A^T W = W^0 + W_i^1(A)$$

Algorithm	Description	# Parameters
MVT1	Probit model without interactions between widgets	$O(ND)$
MVT2	Probit model with interactions between widgets	$O(N^2D^2)$
MVT2c	Probit model with interactions between widgets and between widgets and context	$O(NDGL + N^2D^2)$
N^D-MAB	Non-contextual multi-armed bandit with N^D arms	$O(N^D)$
D-MABs	Independent non-contextual N-armed bandit for each of D widgets	$O(ND)$

Experimental Results

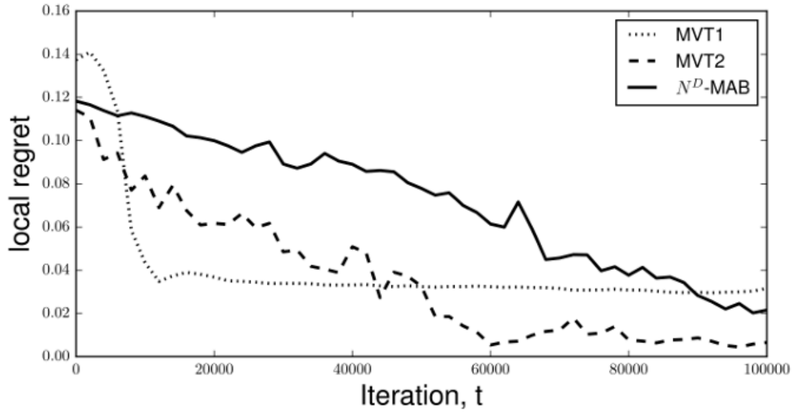


Figure 3: Example run of algorithms on simulated data with $\alpha_1 = 1$, $\alpha_2 = 2$, and $\alpha_c = 0$. Local regret values are averaged over a moving window of 2500 iterations.

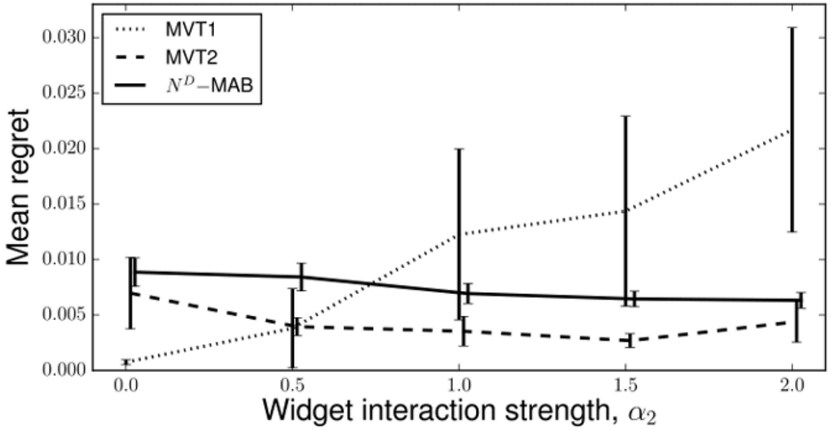


Figure 4: Algorithm performance as α_2 is varied.

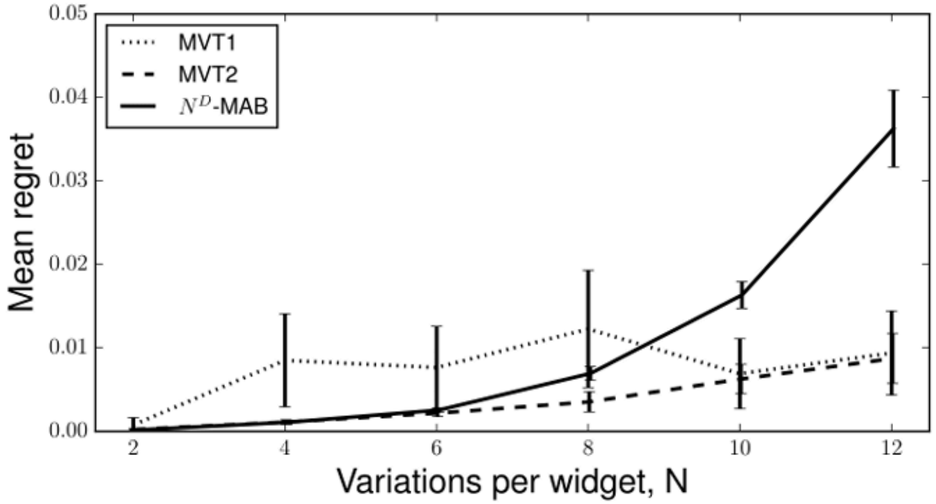


Figure 5: Algorithm performance as N is varied.

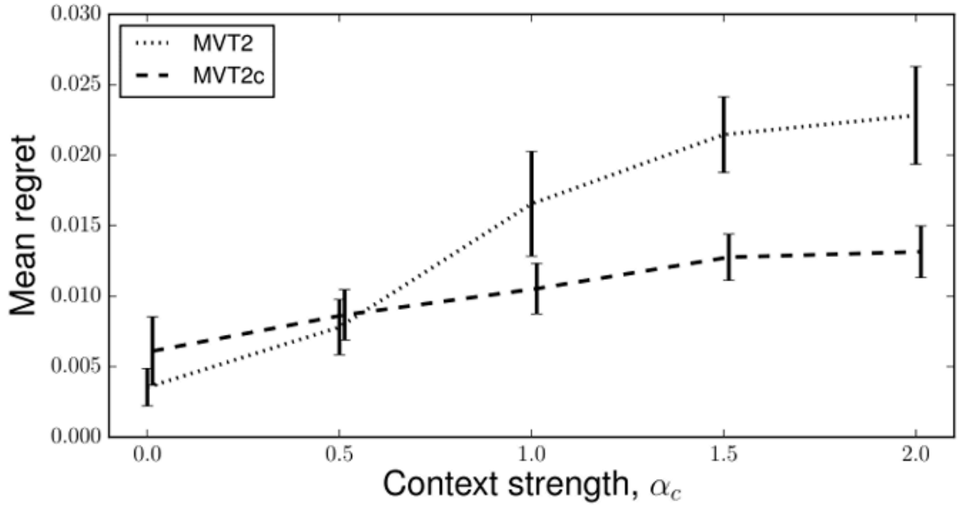


Figure 6: Algorithm performance as α_c is varied.

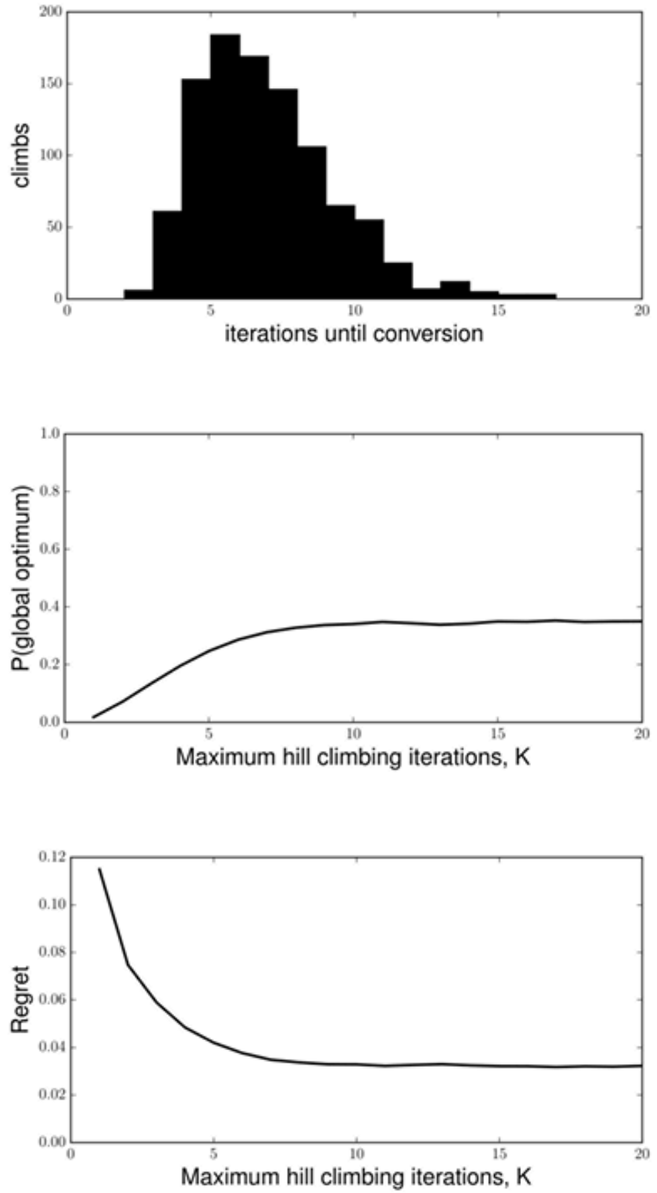


Figure 7: Performance of hill climbing as the number of iterations, K, is varied.

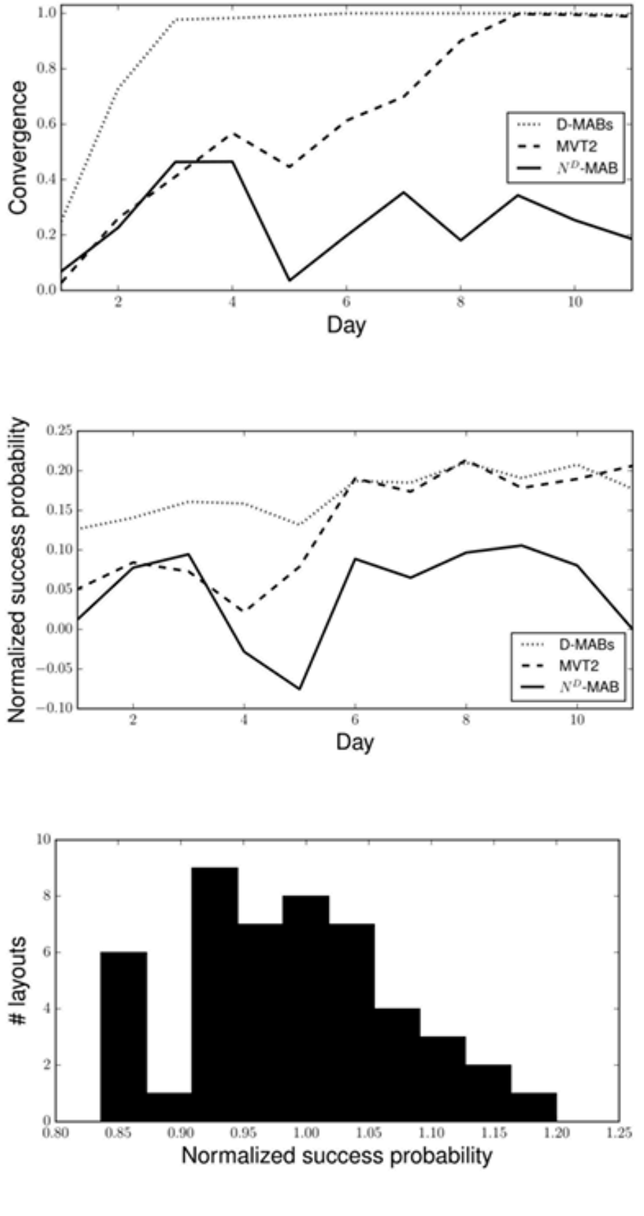


Figure 8: Results of desktop experiment. Top and middle panels show convergence and normalized performance for each algorithm. Bottom panel shows reward of all layouts normalized by median layout.

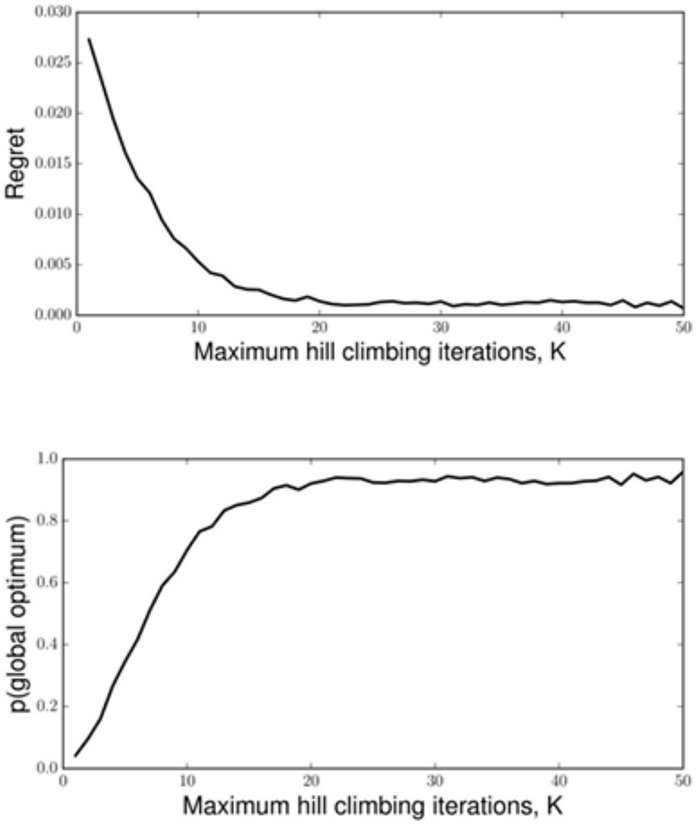


Figure 9: Regret and probability of identifying global optimum in fully trained model as a function of max iterations for hill-climbing.

Conclusion

- Simulation results show that our algorithm scales well to problems involving 1,000s of layouts, converging in a practical amount of time.
- **Scalability**: Our algorithm scales effectively, converging in practical timeframes even with thousands of layouts.
- **Effectiveness**: It captures content interactions and context effects, enabling continuous optimization.
- **Real-world Impact**: Applied to an Amazon service promotion, it delivered significant business impact within a week.
- **Versatility:** Our algorithm isn't domain-specific and can revolutionize decision-making in various contexts.

Swiggy - Contextual Bandits for Ads Recommendations

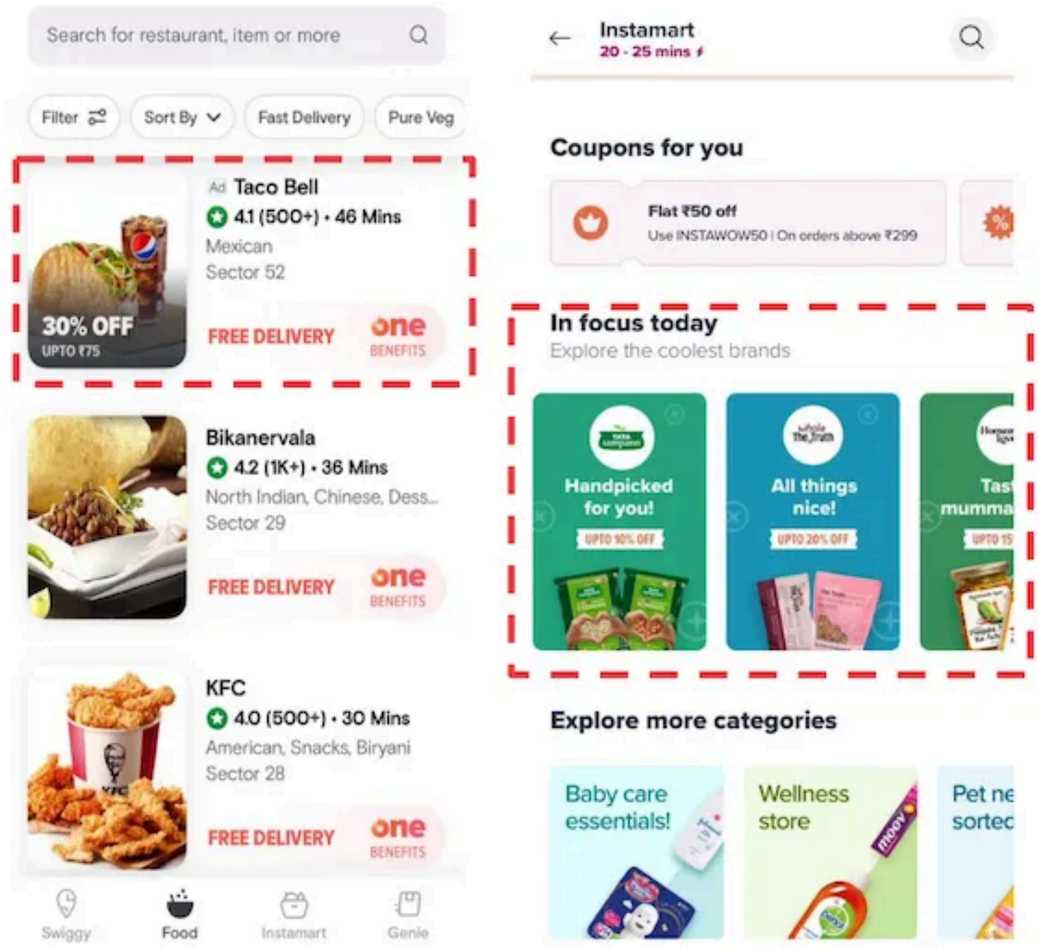
Contextual MAB Setup

- Linear Bandit problem for ad restaurant recommendation.
- Context includes user and restaurant attributes.

Few Definitions :

1. Trial
2. Arm
3. Context
4. Action
5. Reward

Objective: Maximize the total number of clicks in the long run



Swiggy - Contextual Bandits for Ads Recommendations

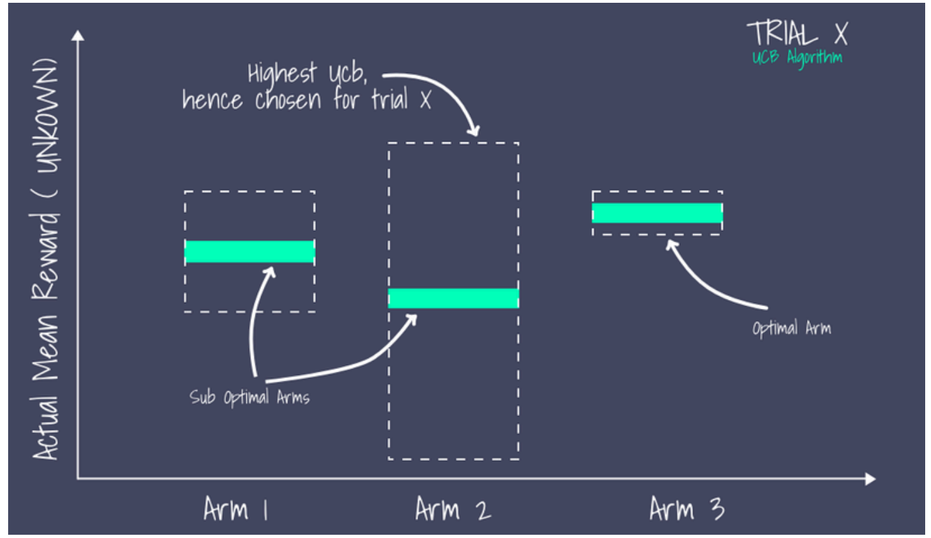
The LinUCB Algorithm

- Scoring Arms

$$a_t \overset{\text{def}}{=} \arg\max_{a \in \mathcal{A}_t} \left(\mathbf{x}_{t,a}^\top \hat{\boldsymbol{\theta}}_a + \alpha \sqrt{\mathbf{x}_{t,a}^\top \mathbf{A}_a^{-1} \mathbf{x}_{t,a}} \right)$$

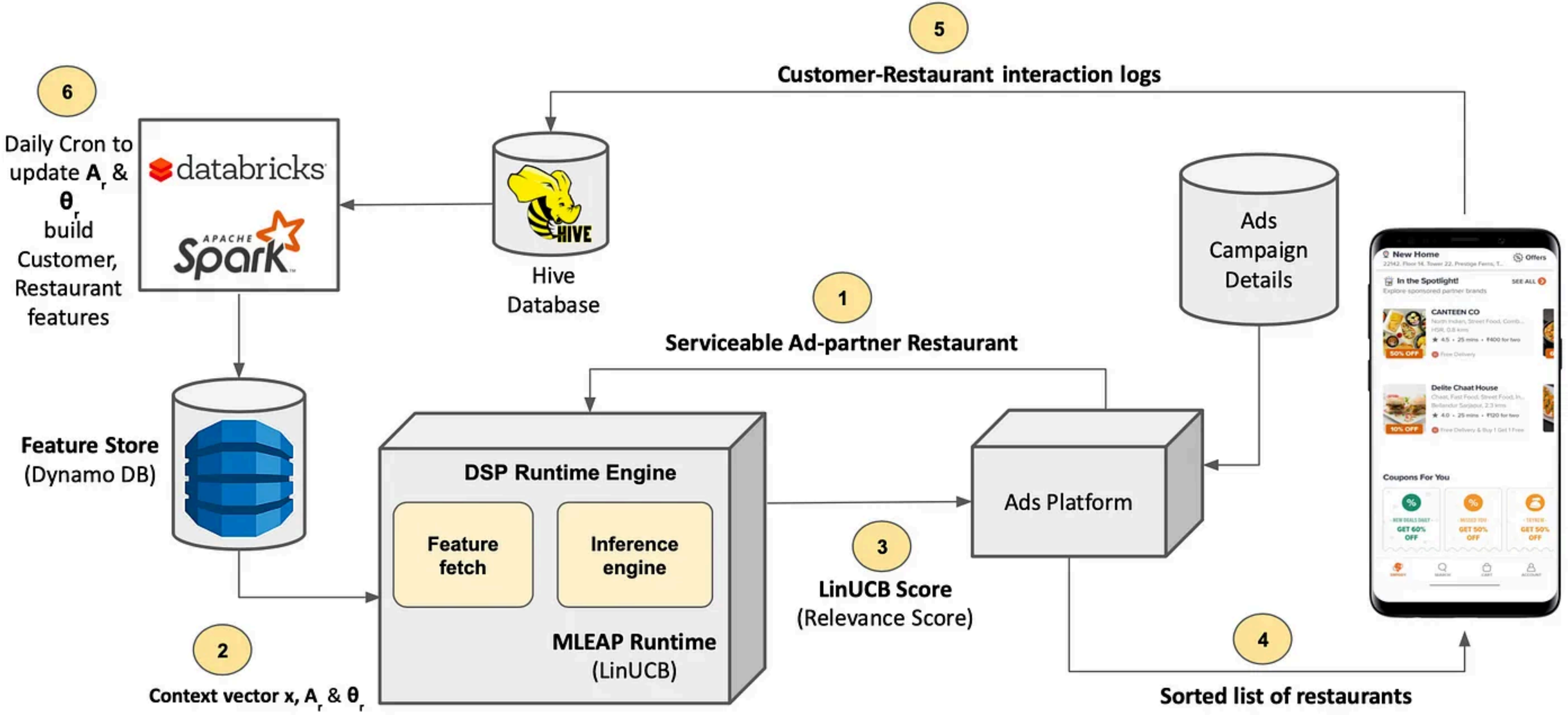
- Updating Model Parameters

$$\mathbf{A}_{a_t} \leftarrow \mathbf{A}_{a_t} + \mathbf{x}_{t,a_t} \mathbf{x}_{t,a_t}^\top$$
$$\mathbf{b}_{a_t} \leftarrow \mathbf{b}_{a_t} + r_t \mathbf{x}_{t,a_t}$$



where,
x (vector) is the Context,
θ(vector) and A(matrix) are model parameters learned for each arm,
α(scalar) is a hyper-parameter which controls the level of exploration
r is the reward (1 or 0)

Ads Recommendation System Architecture



Stochastic bandits for multi-platform budget optimization in online advertising by facebook

- **Problem Statement:**

Optimizing online advertising spending across multiple platforms within a given budget without prior knowledge of each platform's value.

- **Approach**: Formulated as a Stochastic Bandits with Knapsacks problem over T bidding rounds, considering distinct bidding m-tuples representing platforms.
- **Algorithm Extension**: Modified the algorithm proposed by Badanidiyuru et al. [11] to accommodate multiple platforms in both discrete and continuous bid-spaces.
- **Regret Bounds**:

For discrete bids: $O\left(OPT\sqrt{\frac{mn}{B}} + \sqrt{mnOPT}\right)$

For continuous bids: $\tilde{O}\left(m^{1/3} \cdot \min\left\{B^{2/3}, (mT)^{2/3}\right\}\right)$

Stochastic bandits for multi-platform budget optimization in online advertising by facebook

- **Lower Bounds:**
  Discrete case: $\Omega\left(\sqrt{mOPT}\right)$

  Continuous case: $\Omega\left(m^{1/3}B^{2/3}\right)$

- **Real-world Data Analysis**: Demonstrated superior performance of the algorithms over common benchmarks using data from a large internet advertising company.

- **Practical Application**: Algorithms meet necessary criteria for real-world application, providing efficient and effective optimization of online advertising spending across multiple platforms.

References:

[1] A Contextual-Bandit Approach to Personalized News Article Recommendation, Lihong Li, Wei Chu, John Langford, Robert E. Schapire

[2] Daniel N. Hill, Houssam Nassif, Yi Liu, Anand Iyer, S. V. N. Vishwanathan “An Efficient Bandit Algorithm for Realtime Multivariate Optimization” Optimization. In Proceedings of KDD ’17, Halifax, NS, Canada, August 13-17, 2017, 9 pages. https://doi.org/10.1145/3097983.3098184

[3] Swiggy blog on Contextual bandits for ads recommendations

[4] Vashist Avadhanula, Riccardo Colini Baldeschi, Stefano Leonardi, Karthik Abinav Sankararaman, and Okke Schrijvers. "Stochastic bandits for multi-platform budget optimization in online advertising.” WWW '21, page 2805-2817, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450074.

What we plan to do for final review:

- We plan to implement our first paper by Yahoo!
- We plan to implement the LinUCB with Disjoint Linear Models
- Comparing it with other traditional algorithms such as Epsilon Greedy, UCB
- Evaluation using off-policy evaluation method
- Offline evaluation on a dataset which consists about 10,000 user interaction data

Thank you!