Cascading Bandits for Large-Scale Recommendation Problems (ID: 96)

Shi Zong, Hao Ni, Kenny Sung
Carnegie Mellon University

Nan Rosemary Ke
Universite de Montreal

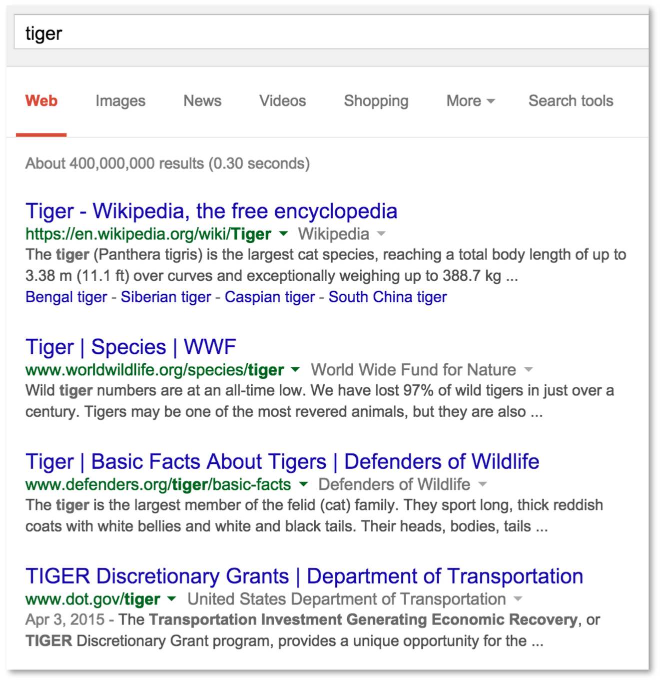
Zheng Wen, Branislav Kveton
Adobe Research

Contributions

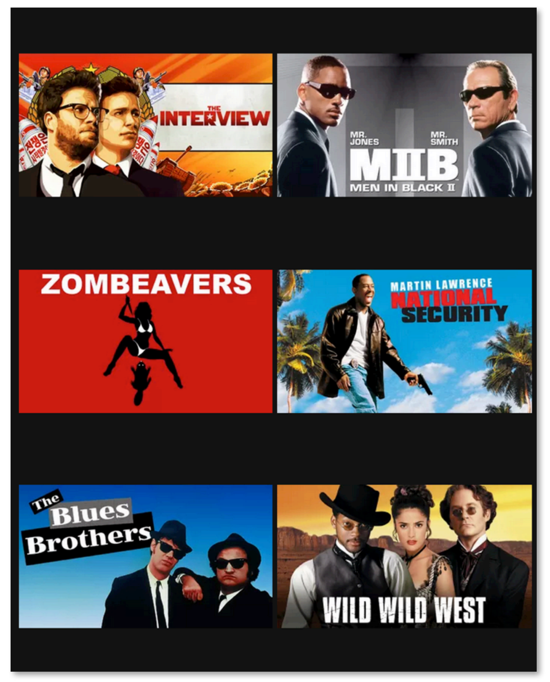
- First work that studies a top-K recommender problem in the bandit setting with cascading feedback and context
  - Assumption: Attraction probabilities are a linear function of the features of items
- Computationally- and statistically-efficient learning algorithms with analysis
- Evaluation on three real-world problems

Motivating Examples

Recommendations



Items are web pages



Items are movies

- Objective: Recommend a list of items that minimizes the probability that none of the recommended items are attractive
- Feedback: Index of the first chosen item which is attractive

Settings

- Ground set E of L items $E = \{1, \dots, L\}$
- Probability distribution P over a binary hypercube $\{0,1\}^E$
- List of K recommended items $A \in \Pi_K(E)$, where $\Pi_K(E)$ is the set of all K-permutations of ground set E

Simplifying Independence Assumption

- Let $P(w) = \prod_{e \in E} \mathrm{Ber}(w(e); \bar{w}(e))$, where $\mathrm{Ber}(\cdot;\theta)$ is a Bernoulli distribution with mean θ. Then

$$\mathbb{E}[f(A, \mathbf{w})] = f(A, \bar{w})$$
$$A^* = \arg\max_{A \in \Pi_K(E)} f(A, \bar{w})$$

Linear Cascading Bandits

CascadeLinTS

Algorithm 1 CascadeLinTS

Inputs: Variance σ^2

// Initialization
$\mathbf{M}_0 \leftarrow I_d$ and $\mathbf{B}_0 \leftarrow \mathbf{0}$

for all $t = 1, \dots, n$ **do**
  $\bar{\theta}_{t-1} \leftarrow \sigma^{-2}\mathbf{M}_{t-1}^{-1}\mathbf{B}_{t-1}$
  $\theta_t \sim \mathcal{N}(\bar{\theta}_{t-1}, \mathbf{M}_{t-1}^{-1})$

  // Recommend a list of K items and get feedback
  for all $k = 1, \dots, K$ **do**
    $\mathbf{a}_k^t \leftarrow \arg\max_{e \in [L] - \{\mathbf{a}_1^t, \dots, \mathbf{a}_{k-1}^t\}} x_e^\mathsf{T}\theta_t$
  $\mathbf{A}_t \leftarrow (\mathbf{a}_1^t, \dots, \mathbf{a}_K^t)$
  Observe click $\mathbf{C}_t \in \{1, \dots, K, \infty\}$
  Update statistics using Algorithm 3

Step 1: Estimate the expected weight of each item e – CascadeLinTS randomly samples parameter vector from a normal distribution

Step 2: Choose the optimal list with respect to estimates

Step 3: Receive feedback and update statistics

CascadeLinUCB

Algorithm 2 CascadeLinUCB

Inputs: Variance σ^2, constant c

// Initialization
$\mathbf{M}_0 \leftarrow I_d$ and $\mathbf{B}_0 \leftarrow \mathbf{0}$

for all $t = 1, \dots, n$ **do**
  $\bar{\theta}_{t-1} \leftarrow \sigma^{-2}\mathbf{M}_{t-1}^{-1}\mathbf{B}_{t-1}$
  for all $e \in E$ **do**
    $\mathbf{U}_t(e) \leftarrow \min\left\{x_e^\mathsf{T}\bar{\theta}_{t-1} + c\sqrt{x_e^\mathsf{T}\mathbf{M}_{t-1}^{-1}x_e}, 1\right\}$

  // Recommend a list of K items and get feedback
  for all $k = 1, \dots, K$ **do**
    $\mathbf{a}_k^t \leftarrow \arg\max_{e \in [L] - \{\mathbf{a}_1^t, \dots, \mathbf{a}_{k-1}^t\}} \mathbf{U}_t(e)$
  $\mathbf{A}_t \leftarrow (\mathbf{a}_1^t, \dots, \mathbf{a}_K^t)$
  Observe click $\mathbf{C}_t \in \{1, \dots, K, \infty\}$
  Update statistics using Algorithm 3

Step 1: Estimate the expected weight of each item e – CascadeLinUCB computes an upper confidence bound for each item

Step 2: Choose the optimal list with respect to estimates

Step 3: Receive feedback and update statistics

Update Statistic

Algorithm 3 Update of statistics in Algorithms 1 and 2

$\mathbf{M}_t \leftarrow \mathbf{M}_{t-1}$
$\mathbf{B}_t \leftarrow \mathbf{B}_{t-1}$
for all $k = 1, \dots, \min\{\mathbf{C}_t, K\}$ **do**
  $e \leftarrow \mathbf{a}_k^t$
  $\mathbf{M}_t \leftarrow \mathbf{M}_t + \sigma^{-2}x_e x_e^\mathsf{T}$
  $\mathbf{B}_t \leftarrow \mathbf{B}_t + x_e \mathbb{1}\{\mathbf{C}_t = k\}$

M_t can be updated incrementally and computationally efficiently in $O(d^2)$ time

- Objective: Minimize the expected cumulative regret in n steps:

$$R(n) = \mathbb{E}\left[\sum_{t=1}^n R(\mathbf{A}_t, \mathbf{w}_t)\right]$$
$$R(\mathbf{A}_t, \mathbf{w}_t) = f(A^*, \mathbf{w}_t) - f(\mathbf{A}_t, \mathbf{w}_t)$$

Analysis

Assumptions: (1) $\hat{w}(e) = x_e^T\theta^*$ for all $e \in E$ (2) $\|x_e\|_2 \le 1$ for all $e \in E$

Theorem: Under the above assumptions, for any $\sigma > 0$ and

$$c \ge \frac{1}{\sigma}\sqrt{d \log\left(1 + \frac{nK}{d\sigma^2}\right) + 2\log(nK)} + \|\theta^*\|_2,$$

if we run CascadeLinUCB with parameters σ and c, then

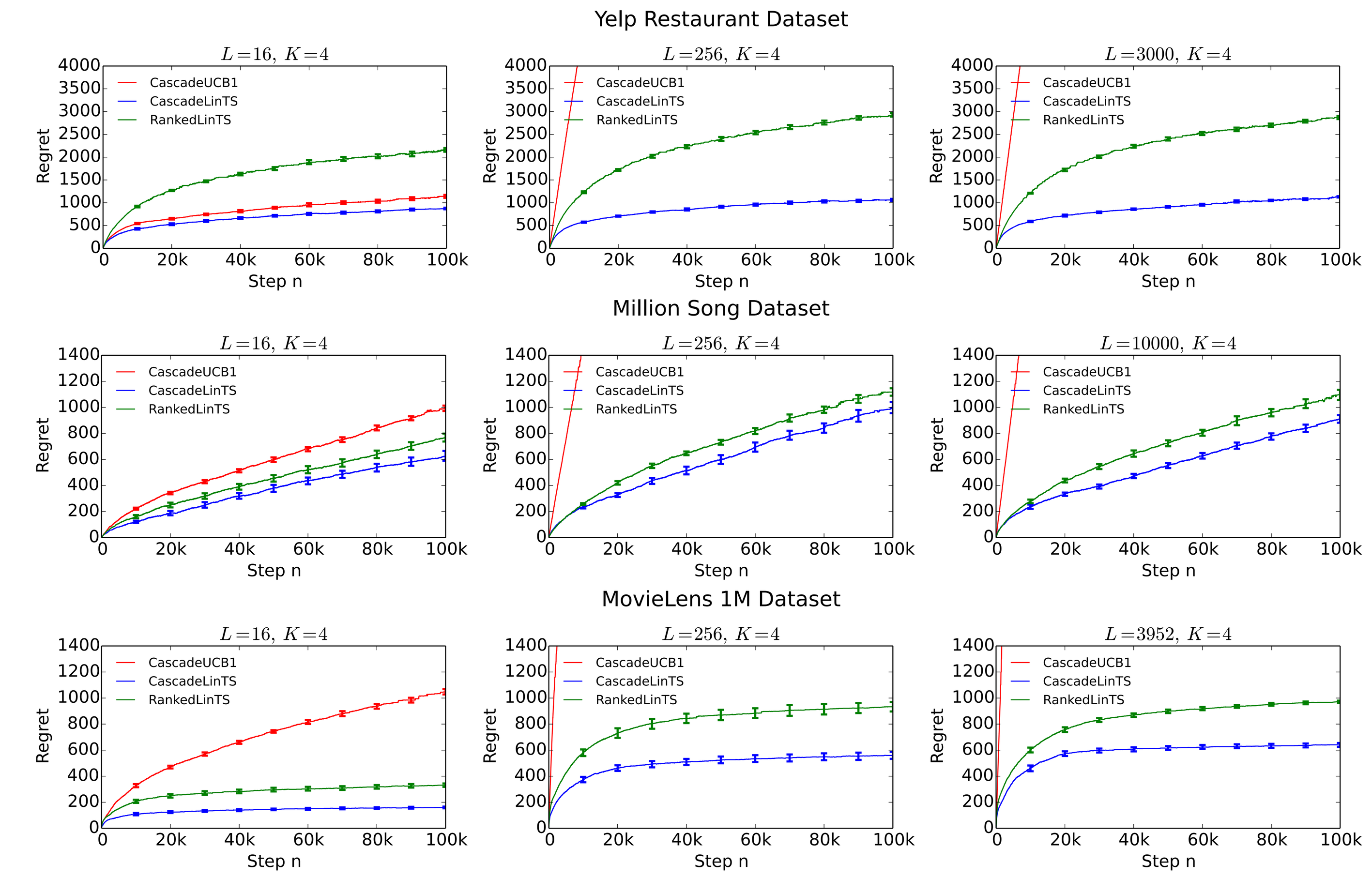
$$R(n) \le 2cK\sqrt{\frac{dn \log\left[1 + \frac{nK}{d\sigma^2}\right]}{\log\left(1 + \frac{1}{\sigma^2}\right)}} + 1.$$

K – Number of recommended items
d -- Number of features
n – Number of iterations

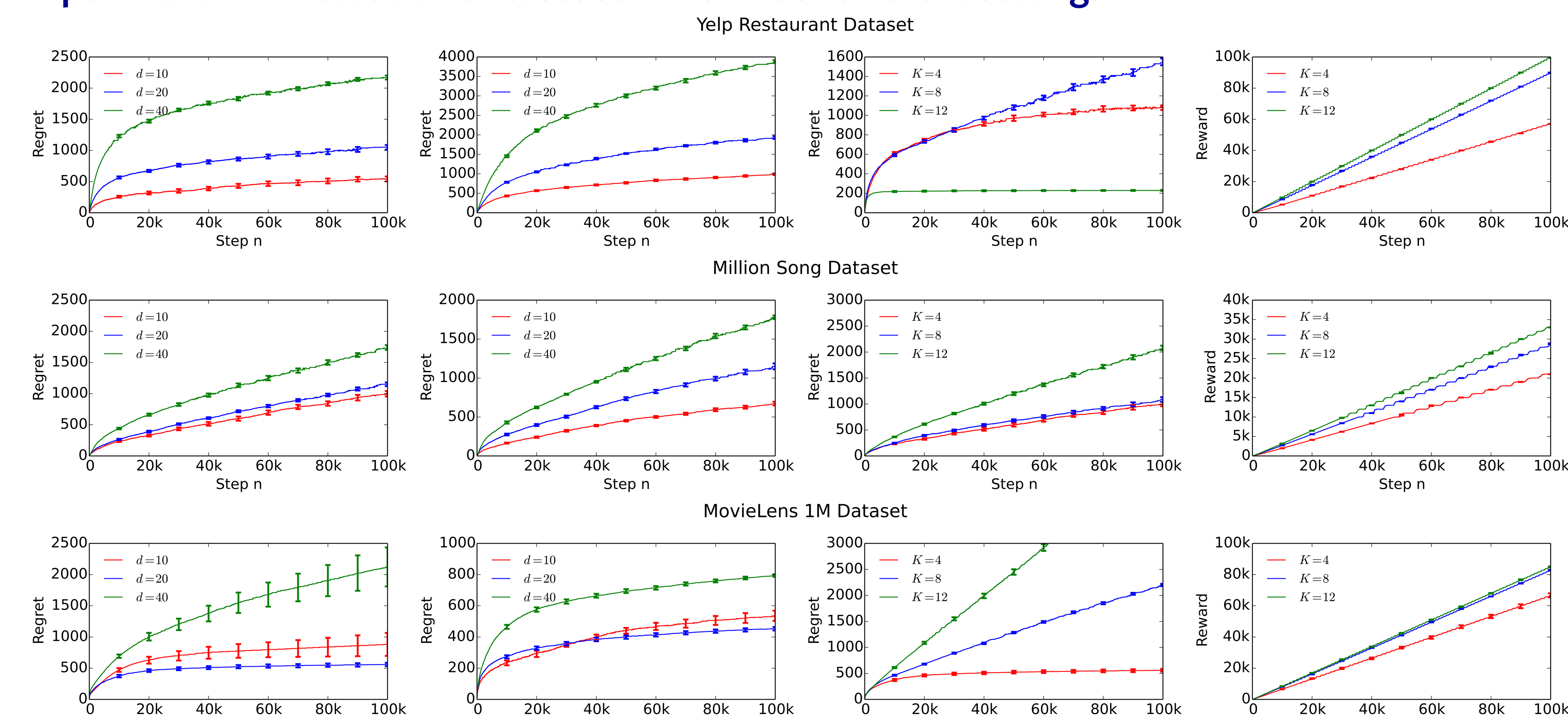
If σ and c are taken some specific values, then the regret could be bounded as
$$R(n) \le \tilde{O}\left(Kd\sqrt{n}\right)$$
Independent on the total number of items L

Experiments

Experiment 1 – Comparisons between our algorithm and existing algorithms



Experiment 2 – Evaluation of CascadeLinTS under different settings



n-step regret for varying number of features d

n-step regret in a subset of each dataset for varying number of features d

n-step regret for varying number of recommended items K

n-step reward for varying number of recommended items K