第40回強化学習勉強会 論文紹介

ICML 2016

DCM Bandits:
Learning to Rank with Multiple Clicks

Authors : Sumeet Katariya, Branislav Kveton, Csaba Szepesva´ri, Zheng Wen

担当者: Koichi Takayama @fullflu

2016/11/9 @グラントウキョウサウスタワー

紹介論文の選択基準

- **問題設定が面白く，かつ理論的にも実験的にも性能保証がある**
  - **問題設定**：Learning to Rankに個人的に興味があった
    - ✳ 検索，メルマガ，推薦などが応用先として期待できて実用性アリ
  - **性能保証**：Banditの証明に個人的に興味があった
    - ✳ 面白そうな問題と組み合わせて，形にできる“力”がほしい
- **~~Deepじゃない~~ ある程度ニッチ**
  - Banditについてはmlp本が非常に詳しいが，Learning to Rankへの適用はまだ扱われていない！
- **おまけ：著者陣の中に，弊勉強会で翻訳執筆中の本の著者が！**
  - “Algorithms for Reinforcement Learning”, Csaba Szepesvári

本発表の位置づけ

- **話すこと**
  - Cascade型のユーザ行動モデルとLearning to Rank問題の気持ち
  - 上記モデルにおけるKL-UCBベースのBanditアルゴリズムの概要
  - リグレット解析の気持ち
  - 実験結果の概要
- **話さないこと**
  - 色々なユーザの行動モデルやLearning to Rank問題の詳細
  - 他（KL-UCB以外）のBanditアルゴリズムの詳細
  - 他の論文（KL-UCBなど）の理論解析の詳細

  ✳ 万が一リクエストがあればざっと書きます…

論文（DCM Bandits）の概要

- **背景：検索システムにおけるユーザの満足度を最大化したい**
  - ユーザの行動モデルについては様々なものが提案されているが，そのほとんどは学習データがすでに揃っていることが前提
  - 適当な行動モデルを仮定したときに，学習データを集めつつユーザの満足度を最大化する方策を考えたい

- **問題設定：Dependent Click Modelの下でのBandit問題**
  - Cascade Modelの下でのBandit問題 [Kveton et al., (2015a)] の拡張

- **提案手法：dcmKL-UCB（KL-UCBの拡張）**
  - 適当な仮定の下で期待リグレットの上限と下限を導出
  - 仮定が満たされるか否かに関わらず，実験性能が良いことを確認
    - ✳ Cascade型複数クリックモデルでは初のregret最適保証つき逐次学習

以降の発表の目次

- **研究背景**
- **問題設定**
- **提案手法**
- **理論解析**
- **実験結果**
- **まとめ，今後の流れ**

※式や図は基本的に紹介論文中のもの，たまに自作 or いらすとや

※色の使い方（曖昧ですが）

青文字　：個人的な解釈などを含む，理解を助けるための強調
赤文字　：強めの一般的な強調，ないしは論文の主張
黒太文字：弱めの一般的な強調

想定する検索システム

ユーザのクエリに対し，サーバがアイテム列を表示する

ユーザ

[クエリ：“Apple”]

[検索結果]

サーバ

想定する検索システム

ユーザのクエリに対し，サーバがアイテム列を表示する

ユーザ

[クエリ：“りんご”]

[検索結果]

サーバ

良いアイテム列を表示したい ⇒ Learning to Rank

Learning to Rank問題の分類

学習のアプローチで大きく二種類に分類される

- **ユーザの行動を明示的にモデル化するもの** **focus**
  - 確率モデルを仮定し，パラメータを推定する
  - 学習時に正しいランキングのデータはなくてもよい
  - 紹介論文はこちらに分類される

    ✳ 解釈が容易，Bandit問題での理論保証を与えやすい（私見）

- **ブラックボックスなランキング予測関数を学習するもの**
  - ランキング学習用の関数とlossを定義し，最適化計算を行う
  - 学習時に(クエリ, 正しいランキング)からなる訓練データが必要
  - 機械学習でよく出てくるのはこちら（RankSVM, Listnetなど…）

    ✳ 解釈が難しい，Bandit問題での理論保証を与えるのが難しい（私見）

ユーザの行動モデル

上から順にアイテムを見て，魅力に思ったものをクリック

ユーザ

[クエリ：“りんご”]

[検索結果]

サーバ

アイテム列

e1は魅力的？

Yes

e1をクリック

⋮

No

e2を見る

⋮

👀

e1
e2
e3
・
・
eL

※見る≠クリック

※魅力でない⇒次を見る

ユーザの行動モデル

上から順にアイテムを見て，魅力に思ったものをクリック

ユーザ

[クエリ：“りんご”]

[検索結果]

サーバ

アイテム列

e1は魅力的？

Yes → e1をクリック

No → e2は魅力的？

Yes → e2をクリック

⋮

No → e3を見る

⋮

👀

~~e1~~
e2
e3
・
・
eL

ユーザの行動モデル

上から順にアイテムを見て，魅力に思ったものをクリック

ユーザ

[クエリ：“りんご”] →

サーバ

← [検索結果]

アイテム列

~~e1~~
e2
e3
・
・
eL

e1は魅力的？
- Yes → e1をクリック
- No → e2は魅力的？
  - Yes → e2をクリック
    ⋮
  - No → e3を見る
    ⋮

* linear traversal hypothesis
** examination hypothesis

以降でもこれらを仮定

ユーザの行動モデル：Cascade Model

Craswell et al., (2008)

一度クリックしたら二度と検索に戻らない

ユーザ

[クエリ：“りんご”] →

← [検索結果]

サーバ

アイテム列

e1は魅力的？ →

Yes → e1をクリック

No → e2を見る

e1
e2
e3
・
・
eL

※クリック=満足

ユーザの行動モデル：Cascade Model

Craswell et al., (2008)

一度クリックしたら二度と検索に戻らない

ユーザ

[クエリ：“りんご”]

[検索結果]

サーバ

アイテム列

e1は魅力的？

Yes → e1をクリック

No → e2は魅力的？

Yes → **e2をクリック**

No → e3を見る

⇒ 検索終了（e2に満足したと解釈)

~~e1~~
e2
e3
・
・
eL

左の図がCascade型！

※複数回クリックを扱えない
　ので制限が強いモデル

ユーザの行動モデル：Dependent Click Model

Guo et al., (2009b)

クリックしたアイテムに満足しなければ検索に戻る

ユーザ

[クエリ：“りんご”]

[検索結果]

サーバ

アイテム列

e1は魅力的？

Yes

e1をクリック

No

e2を見る

e1
e2
e3
・
・
eL

※クリック≠満足

ユーザの行動モデル：Dependent Click Model

Guo et al., (2009b)

クリックしたアイテムに満足しなければ検索に戻る

ユーザ

[クエリ：“りんご”]

[検索結果]

サーバ

アイテム列

e1は魅力的？

Yes → e1に満足した？

No → e2を見る

e1に満足した？ No → e2を見る

e1に満足した？ Yes → **検索終了（e1に満足したと解釈）**

e1
e2
e3
・
・
eL

ユーザの行動モデル：Dependent Click Model

Guo et al., (2009b)

クリックしたアイテムに満足しなければ検索に戻る

ユーザ

[クエリ：“りんご”]

[検索結果]

サーバ

アイテム列

e1は魅力的？

Yes

No

e1に満足した？

e2を見る

No

Yes

検索終了（e1

e1
e2
e3
・
・
eL

クリックの依存関係をモデル化

※複数回クリックを扱えるので,
Cascade Modelよりは現実的

DCMのイメージ（紹介論文のFigure 1）

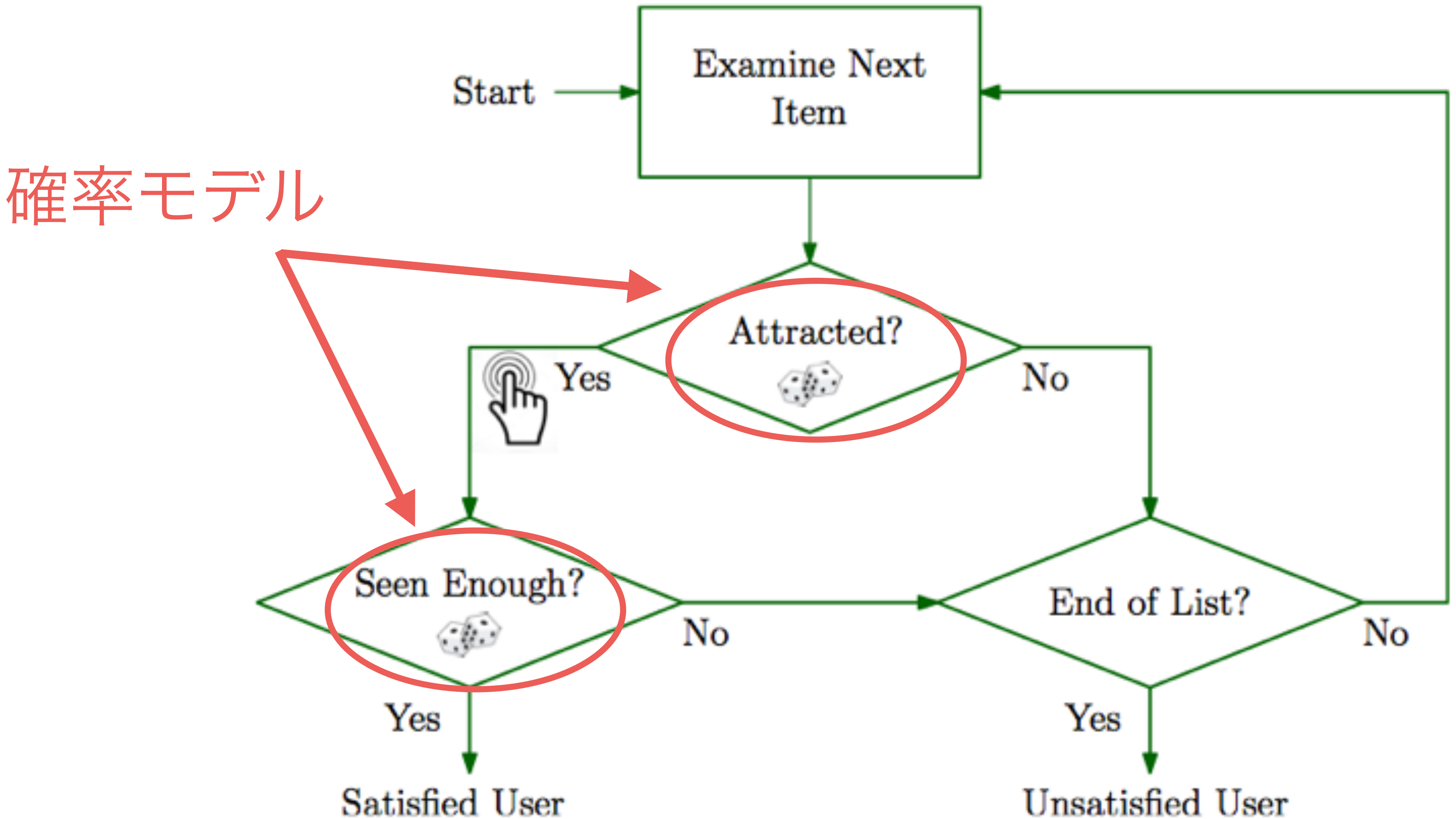
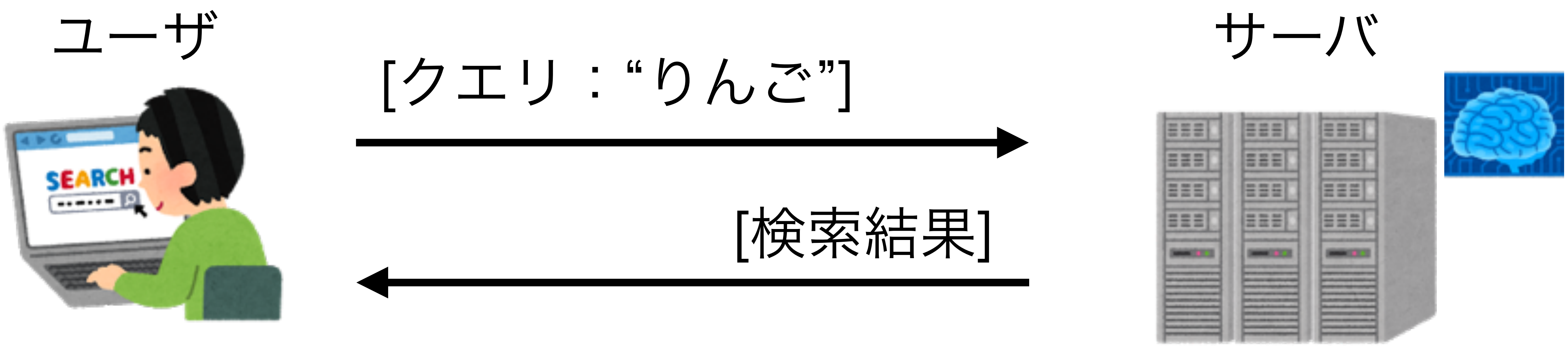


Figure 1. Interaction between the user and items in the DCM.

DCM : Dependent Click Modelの略

サーバのモデル

ユーザのクエリに対し，満足度が高いアイテム列を出したい



サーバのタスク

- **(クエリ, アイテム列, クリックログ)の三つ組を蓄積する**
- **ユーザの行動モデルを学習する**
- **学習した行動モデルに基づき満足度が高いアイテム列を出す**

以降，サーバはクリックログ以上のデータ（e.g., 購買ログ）を取得できないと仮定
※現実と比べると，少し厳しい仮定な気もするが…

探索と活用のトレードオフ

“データが多くなるのを 待ってはいられない”

～Cold Start問題～

- **探索**：色々なパターンのアイテム列を出してデータを収集する
  - 収集中は魅力的でないアイテムを出してしまう恐れがある
- **活用**：各時点で魅力的と思われるアイテム列を出す
  - 探索が不十分だと，魅力的なアイテムを出さない恐れがある
  - 「本当は魅力的なアイテムを，魅力的でないと思い込んでしまう」

➡ **探索と活用をバランスよく行い，満足度を最大化したい**
（バンディット問題を定式化して，解きたい）

Bandit問題

本田 & 中村, (2016)

詳しい議論はMLP本が詳しいのでそちらに任せる

エージェント（今回だとサーバ）の振る舞いの概要

1. 初めにアイテム列を選ぶ
2. 選んだアイテム列に対応する報酬に関連する信号*を環境から得る
   - 今回は報酬が確率的に得られるケースのみを扱う
3. 長期的な累積報酬の最大化（＝リグレットの最小化）を目的として，過去に選んだアイテム列と信号を用いて次のアイテム列を選ぶ
4. 2〜3を繰り返す

*報酬を一般化したものが報酬に関連する信号

- **以降，エージェントはクエリごとに別々に学習を行う場合を考える**
  - つまり，いわゆるContextual Banditではない
- **よって，一つのクエリのみに着目した議論を進める**

紹介論文の位置づけ

ユーザの行動モデルの学習

	単一クリック	複数クリック
非Cascade型 （クリック依存なし）	クリック率予測（by 二値分類？）	
Cascade型 （クリック依存あり）	**Cascade model** [Craswell et al.,(2008)]	**DCM, UBM, CCM, DBN** [参考文献にまとめた]

Bandit問題

	単一クリック	複数クリック
非Cascade型 （クリック依存なし）	よくあるBandit	・**Ranked Bandit** [Slivkin et al., (2013)] ・**複数選択Bandit** [Komiyama et al., (2015)]
Cascade型 （クリック依存あり）	**Cascading Bandits** [Kveton et al.,（2015)]	**DCM Bandits** **[紹介論文]**

著者が同一グループ

問題設定

- 研究背景
- 問題設定
- 提案手法
- 理論解析
- 実験結果
- まとめ，今後の流れ

【再掲】DCM（紹介論文のFigure 1）

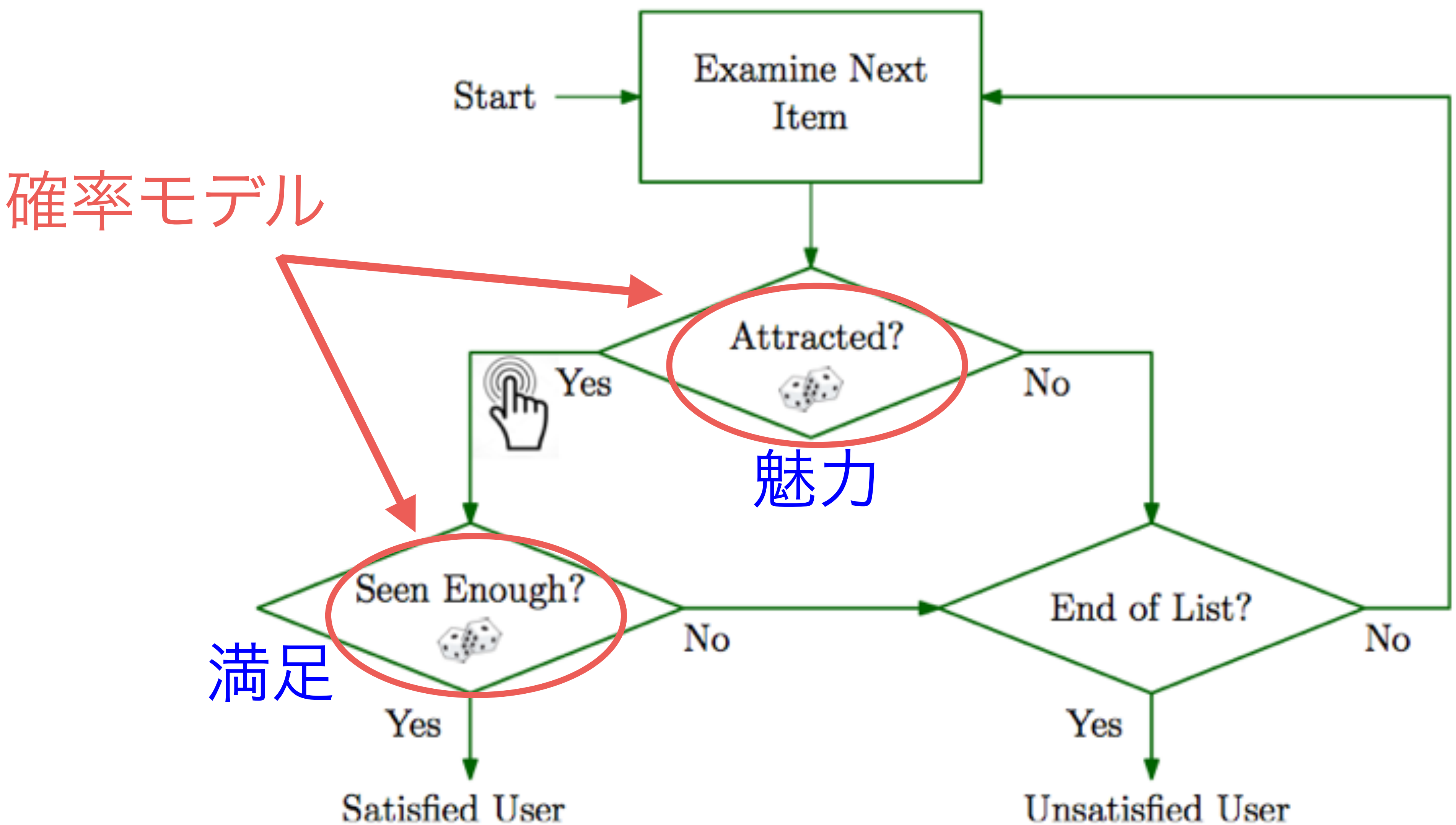


Figure 1. Interaction between the user and items in the DCM.

DCM : Dependent Click Modelの略

問題設定：DCM（事前設定）

サーバが知っているもの

L ：**アイテムの総数**

$E = [L]$ ：**アイテムの集合**

K ：**検索結果に含まれるアイテム数（場所の数）**

$\Pi_K(E) \subset E^K$ ：**L個のアイテムからK個を選び並べた全順列**

n ：**クエリが投げられる総数（セッション数）**

サーバが知らないもの

P_{w} ：$\{0,1\}^E$ **上の確率分布（アイテムの魅力分布）**

P_{v} ：$\{0,1\}^K$ **上の確率分布（場所の満足されやすさ分布）**

- それぞれL次元, K次元の単位超立方体の頂点上の確率分布

$\bar{w} \in [0,1]^E$ ：**アイテムが魅力的である確率*の配列**

- attractive probabilities

$\bar{v} \in [0,1]^K$ ：**クリックされた場所でユーザが満足する確率の配列**

- termination probabilities

*アイテムが魅力的である確率 = それが見られたときに，クリックされる確率

問題設定：DCM（運用後に変わるもの）

サーバが知っているもの $t \in [n]$

$\mathbf{A}_t = (\mathbf{a}_1^t, \ldots, \mathbf{a}_K^t) \in \Pi_K(E)$: **時刻 t でサーバが選ぶアイテム列**

$\mathbf{c}_t \in \{0,1\}^K$: **各時刻で各場所がクリックされるか否かの確率変数**

サーバが一部を知っているもの（最終クリック場所までの情報は知っている）

$\mathbf{w}_t \sim P_{\mathrm{w}}$: **各時刻で各アイテムが魅力的か否かの確率変数**

- 各時刻のサンプリングは独立で，ドメインは $\{0,1\}^E$

$\mathbf{v}_t \sim P_{\mathrm{v}}$: **各時刻で各場所がクリック後に満足されるか否かの確率変数**

- 各時刻のサンプリングは独立で，ドメインは $\{0,1\}^K$

サーバが知らないもの

$\mathrm{r}_t \in \{0,1\}$: **各時刻でユーザーが満足したか否かの報酬を表す確率変数**

- 選ばれたK個のアイテムの中で，少なくとも一つに満足すれば1
- どのアイテムにも満足しなければ0

問題設定：DCM（確率分布の仮定）

確率分布の独立性の仮定

$$P_{\mathrm{w}}(w) = \prod_{e \in E} \mathrm{Ber}(w(e); \bar{w}(e))$$

$$P_{\mathrm{v}}(v) = \prod_{k \in [K]} \mathrm{Ber}(v(k); \bar{v}(k))$$

for any $w \in \{0,1\}^E$ and $v \in \{0,1\}^K$

- 各アイテムの魅力の有無は独立で，パラメータ $\bar{w}(e)$ のベルヌーイ分布に従う
- 各場所の満足度の有無は独立で，パラメータ $\bar{v}(k)$ のベルヌーイ分布に従う

※indexは下付きではなく括弧で表記：w(e)は配列wのindex eの値

問題設定：Reward function

Reward function f を以下のように定義

$$f(A, w, v) = 1 - \prod_{k=1}^{K}(1 - v(k)w(a_k))$$

$$f : \Pi_K(E) \times [0,1]^E \times [0,1]^K \to [0,1]$$

Reward functionの定義の妥当性

- ω, v が{0,1}からなるベクトルのとき，先に定義した報酬を返す

- 真の確率配列 $\bar{w}$, $\bar{v}$ を代入すると，アイテムAに満足する確率を返す

$$f(A, \bar{w}, \bar{v}) = 1 - \prod_{k=1}^{K}(1 - \underline{\bar{v}(k)\bar{w}(a_k)})$$

k番目のアイテムを見たときに満足する確率

➡ **アイテムの満足度を測る指標として整合性がある**

問題設定：リグレット

期待リグレットR(n)を以下のように定義

"期待リグレット"の一般的な定義（論文にはない式）

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

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

$$A^* = \arg\max_{A\in\Pi_K(E)} f(A,\bar{w},\bar{v})$$

確率分布の独立性の仮定から f の期待値を以下のように計算できることを使った

- f の期待値はアイテム列 $A=(a_1,\ldots,a_K)$ に満足する確率と一致する

$$\mathbb{E}[f(A,\mathbf{w},\mathbf{v})] = 1 - \prod_{k=1}^{K}(1-\mathbb{E}[\mathbf{v}(k)]\,\mathbb{E}[\mathbf{w}(a_k)])$$

$$= 1 - \prod_{k=1}^{K}(1-\bar{v}(k)\bar{w}(a_k))$$

$$= f(A,\bar{w},\bar{v})$$

Partial monitoring問題の難しさ

今回考えるモデルでは，報酬を直接観測できない！

・例として，あるクエリに対して以下のログが得られた場合を考える
- サーバが選んだアイテム列： $\mathbf{A}_t = (1, 2, 3, 4)$
- それに対するクリックログ： $\mathbf{c}_t = (0, 1, 1, 0)$

以下の2つのユーザ行動を区別できないため，学習が難しい

報酬1：アイテム3に満足した
- このとき，4つ目のアイテムは見られないので，クリックされない

報酬0：アイテム3に満足せずアイテム4を見たがクリックしなかった

報酬が観測できない代わりに，報酬に関連する信号（今回だとクリック）を観測できるような問題を **Partial monitoring問題** と呼ぶ

Partial monitoring問題へのアプローチ

仮定を一つ追加することで効率的に解ける

- そのまま一般的な手法を使用するのは適当ではない
  - 「サーバのactionの候補がKに対して指数関数的に増える」ためと論文には書かれている
  - 計算量的な問題と学習効率の問題がともにありそう[要出典]

- **追加する仮定**：$\bar{v}$ の**順序**が分かっているという仮定
  - $\bar{v}(1) \geq \ldots \geq \bar{v}(K)$ としても一般性は失われない
  - **同じ順序であれば，どんな値でも最適なアイテム列は変わらない！**

賢い！

$$^{\forall}\tilde{v} \in [0,1]^K \text{ s.t. } \tilde{v}(1) \geq \ldots \geq \tilde{v}(k)$$

$$A^* = \arg\max_{A \in \Pi_K(E)} f(A, \bar{w}, \tilde{v})$$

➡ **$\bar{v}$ の推定が不要になるため，パラメータが減って効率的に！**

保存するクリックデータ

最後にクリックしたアイテムまでを保存

・最後のクリック以降のユーザの行動は観測できない [前述]

➡ **各時刻で $\min\left\{\mathbf{C}_t^{\mathrm{last}}, K\right\}$ までのアイテムのログをDBに保存**

$$\mathbf{C}_t^{\mathrm{last}} = \max\left\{k \in [K] : \mathbf{c}_t(k) = 1\right\} \quad \text{, where } \max \emptyset = +\infty$$

問題設定

- 研究背景
- 問題設定
- 提案手法
- 理論解析
- 実験結果
- まとめ，今後の流れ

提案アルゴリズム：dcmKL-UCB

各時刻で関数fの値が最大のアイテムK個を選ぶ

Algorithm 1 dcmKL-UCB for solving DCM bandits.

```
// Initialization
Observe w_0 ~ P_w
∀e ∈ E : T_0(e) ← 1
∀e ∈ E : ŵ_1(e) ← w_0(e)

for all t = 1, ..., n do
  for all e = 1, ..., L do
    Compute UCB U_t(e) using (1)

  // Recommend and observe
  A_t ← arg max_{A∈Π_K(E)} f(A, U_t, ṽ)
  Recommend A_t and observe clicks c_t ∈ {0,1}^K
  C_t^last ← max {k ∈ [K] : c_t(k) = 1}

  // Update statistics
  ∀e ∈ E : T_t(e) ← T_{t-1}(e)
  for all k = 1, ..., min {C_t^last, K} do
    e ← a_k^t
    T_t(e) ← T_t(e) + 1
    ŵ_{T_t(e)}(e) ← (T_{t-1}(e) ŵ_{T_{t-1}(e)}(e) + c_t(k)) / T_t(e)
```

紹介論文Algorithm 1

初期化

事前にセッションをL回行い各アイテムを1位に出す

$\mathbf{T}_t(e)$: 時刻tまでにアイテムeがDBに記録された回数

// Initialization
$\boxed{\text{Observe } \mathbf{w}_0 \sim P_{\mathrm{w}}}$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

$\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e)$: アイテムeがDBに $\mathbf{T}_t(e)$ 回記録された時点での $\bar{w}(e)$ の推定値

Q : 分布 Pw を知らないサーバがそこからのサンプルを得るには？

A : 1位に出したアイテムは絶対に見られるので，アイテム数だけ事前にセッションを行い各アイテムを1位に出せばよい

（eが**クリックされたら** $\mathbf{w}_0(e) = 1$, **されなければ** $\mathbf{w}_0(e) = 0$）

UCB（Upper Confidence Bound）計算

各アイテムについて推定値の信頼度の上限を(1)式で求める

$$\mathbf{U}_t(e) = \max\{q \in [w, 1] : w = \hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e)\,,$$
$$\underline{\mathbf{T}_{t-1}(e)}\,\underline{D_{\mathrm{KL}}(w \,\|\, q)} \le \log t + 3 \log\log t\}\,, \qquad (1)$$

KL divergence

DB内のアイテム数が少ないほど上限は大きめに
（不確かなときは楽観的に）

Algorithm 1 dcmKL-UCB for solving DCM bandits.

// Initialization
Observe $\mathbf{w}_0 \sim P_{\mathrm{w}}$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for all $t = 1, \dots, n$ **do**
  for all $e = 1, \dots, L$ **do**
    $\boxed{\text{Compute UCB } \mathbf{U}_t(e) \text{ using (1)}}$

  // Recommend and observe
  $\mathbf{A}_t \leftarrow \arg\max_{A \in \Pi_K(E)} f(A, \mathbf{U}_t, \tilde{v})$
  Recommend $\mathbf{A}_t$ and observe clicks $\mathbf{c}_t \in \{0, 1\}^K$
  $\mathbf{C}_t^{\mathrm{last}} \leftarrow \max\{k \in [K] : \mathbf{c}_t(k) = 1\}$

  // Update statistics
  $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
  for all $k = 1, \dots, \min\{\mathbf{C}_t^{\mathrm{last}}, K\}$ **do**
    $e \leftarrow \mathbf{a}_k^t$
    $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
    $\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) \leftarrow \dfrac{\mathbf{T}_{t-1}(e)\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbf{c}_t(k)}{\mathbf{T}_t(e)}$

$D_{\mathrm{KL}}(w \,\|\, q)$ は $q \ge w$ において単調増加するのでUCBは容易に計算可能

ユーザとサーバのやり取り

サーバはアイテム列を出し，ユーザからクリックログを得る

vの順序とUCBの順序が合うように選ぶだけでよいはず

⇒ **実際にはfの値は求めなくてよい？[要確認]**

$$f(A, w, v) = 1 - \prod_{k=1}^{K} (1 - v(k) w(a_k))$$

Algorithm 1 dcmKL-UCB for solving DCM bandits

// Initialization
Observe $\mathbf{w}_0 \sim P_\mathrm{w}$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for all $t = 1, \dots, n$ **do**
  for all $e = 1, \dots, L$ **do**
    Compute UCB $\mathbf{U}_t(e)$ using (1)

  // Recommend and observe
  $\mathbf{A}_t \leftarrow \arg\max_{A \in \Pi_K(E)} f(A, \mathbf{U}_t, \tilde{v})$
  Recommend $\mathbf{A}_t$ and observe clicks $\mathbf{c}_t \in \{0, 1\}^K$
  $\mathbf{C}_t^{\mathrm{last}} \leftarrow \max\{k \in [K] : \mathbf{c}_t(k) = 1\}$

  // Update statistics
  $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
  for all $k = 1, \dots, \min\{\mathbf{C}_t^{\mathrm{last}}, K\}$ **do**
    $e \leftarrow \mathbf{a}_k^t$
    $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
    $\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) \leftarrow \dfrac{\mathbf{T}_{t-1}(e) \hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbf{c}_t(k)}{\mathbf{T}_t(e)}$

推定値の更新（論文では詳細説明無し）

最尤推定量のように見える

$p_t(e)$: 時刻tまでにeがクリックされた数

$q_t(e)$: 時刻tまでにeがクリックされなかった数

$$p_t(e) + q_t(e) = T_t(e)$$

Algorithm 1 dcmKL-UCB for solving DCM bandits.

// Initialization
Observe $\mathbf{w}_0 \sim P_{\mathrm{w}}$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for all $t = 1, \dots, n$ **do**
  for all $e = 1, \dots, L$ **do**
    Compute UCB $\mathbf{U}_t(e)$ using (1)

  // Recommend and observe
  $\mathbf{A}_t \leftarrow \arg\max_{A \in \Pi_K(E)} f(A, \mathbf{U}_t, \tilde{v})$
  Recommend $\mathbf{A}_t$ and observe clicks $\mathbf{c}_t \in \{0,1\}^K$
  $\mathbf{C}_t^{\mathrm{last}} \leftarrow \max\{k \in [K] : \mathbf{c}_t(k) = 1\}$

  // Update statistics
  $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
  for all $k = 1, \dots, \min\{\mathbf{C}_t^{\mathrm{last}}, K\}$ **do**
    $e \leftarrow \mathbf{a}_k^t$
    $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
    $\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) \leftarrow \dfrac{\mathbf{T}_{t-1}(e)\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbf{c}_t(k)}{\mathbf{T}_t(e)}$

最尤推定量の計算

$$\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) = \frac{p_t(e)}{\mathbf{T}_t(e)} = \frac{p_{t-1}(e) + \mathbf{c}_t(k)}{\mathbf{T}_t(e)} = \frac{\mathbf{T}_{t-1}(e)\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbf{c}_t(k)}{\mathbf{T}_t(e)}$$

完全に一致

問題設定

- 研究背景
- 問題設定
- 提案手法
- 理論解析
- 実験結果
- まとめ，今後の流れ

理論解析の結果（気持ち）

Lに対して線形で悪化し，nに対してlogで悪化し，Kとは逆相関

- $\bar{v}(1) \geq \ldots \geq \bar{v}(K)$ のとき（上限のみ）

$$O\Big(\gamma(L-K)\log(n)\Big)$$

- $\bar{v}(1) = \ldots = \bar{v}(K) = \gamma$ のとき

上限

$$O\Big(\gamma(L-g(K))\log(n)\Big)$$

gはKの増加正値関数？

下限

$$\Omega\left(\gamma(L-K)\frac{\Delta}{D_{\mathrm{KL}}(p-\Delta \,\|\, p)}\log n\right) \quad \forall \textit{DCM bandit } B_{\mathrm{LB}}$$

バウンドを示すときの気持ち

上限

提案アルゴリズムは,
考えているクラスに含まれるあらゆる問題に対して,
期待値の意味で “それ以上悪くならない”

下限

強一致性を持つあらゆるアルゴリズムは,
考えているクラスに含まれるあらゆる問題に対して,
期待値の意味で“それ以上良くならない”

※以降の証明部分については，読みたくない人は実験まで飛んでもよいが,
notationだけは読んでおいた方が実験設定が分かりやすいかも

理論解析：以下の3つのバウンドを示す

任意の $\epsilon > 0$ に対して，ある正値関数 $C_2(\epsilon)$, $\beta(\epsilon)$ が存在して…

- $\bar{v}(1) \geq \ldots \geq \bar{v}(K)$ のとき（上限のみ）

$$R(n) \leq (1+\varepsilon) \sum_{i=1}^{K} \frac{\bar{v}(i) - \bar{v}(i+1)}{\alpha} \times \sum_{e=i+1}^{L} \frac{\Delta_{e,i}(1+\log(1/\Delta_{e,i}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(i))} (\log n + 3 \log\log n) + C,$$

$$\textit{where } \bar{v}(K+1) = 0, \;\; C = \textstyle\sum_{i=1}^{K} \frac{\bar{v}(i)-\bar{v}(i+1)}{\alpha} \left(iL \frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7i \log\log n \right)$$

- $\bar{v}(1) = \ldots = \bar{v}(K) = \gamma$ のとき

上限

$$R(n) \leq \frac{\gamma}{\alpha} \sum_{e=K+1}^{L} \frac{(1+\varepsilon)\Delta_{e,K}(1+\log(1/\Delta_{e,K}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(K))} \times (\log n + 3 \log\log n) + C,$$

$$\textit{where } C = \tfrac{\gamma}{\alpha} \left(KL \frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7K \log\log n \right)$$

下限

$$\liminf_{n \to \infty} \frac{R(n)}{\log n} \geq \gamma\alpha \frac{(L-K)\Delta}{D_{\mathrm{KL}}(p - \Delta \,\|\, p)} \qquad {}^{\forall}\textit{DCM bandit } B_{\mathrm{LB}}$$

準備：Cascade期待リグレット

- **Cascade reward を定義** $i \in [K]$

$$f_i(A, w) = 1 - \prod_{k=1}^{i}(1 - w(a_k))$$

名前の由来：Cascade modelの下でアイテムを i 個出したときのrewardと一致
※Cascade model以外のモデルを考える場合もこのrewardは評価できる

- **Cascade rewardに対応するCascade期待リグレットを定義**

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

論文中では *expected cumulative cascade regret* と呼ばれる

命題1：Cascade期待リグレットの上限

- アイテム $e^* \in A^*$ と $e \notin A^*$ の魅力度のgapを以下のように表記

$$\Delta_{e,e^*} = \bar{w}(e^*) - \bar{w}(e)$$

- $\bar{w}(1) \geq \ldots \geq \bar{w}(L)$ と仮定しても一般性を失わない

命題1：dcmKL-UCBアルゴリズムを用いたとき

任意の $i \in [K]$ と $\epsilon > 0$ に対して，ある正値関数 $C_2(\epsilon)$, $\beta(\epsilon)$ が存在し，

$$R_i(n) \leq \sum_{e=i+1}^{L} \frac{(1+\varepsilon)\Delta_{e,i}(1+\log(1/\Delta_{e,i}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(i))} \times (\log n + 3\log\log n) + C\,,$$

$$where \quad C = iL\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7i\log\log n$$

証明：Kveton et al. (2015a)の上限の証明を借りることが可能

- データが同じとき，dcmKL-UCBはCascade KL-UCBと同じ
- $^\forall \mathbf{A}_t, \mathbf{w}_t$; DBに入れるデータ数はdcmKL-UCB $\geq$ Cascade KL−UCB

※紹介論文での証明はこれだけ…ひとまずこの命題は成り立つと思って下さい（要考察）

準備：2つの簡単な補題

- 以下のような表記を定義すると，2つの補題が言える

$$p_{\max} = \arg\max_{e \in [L]} \bar{w}(e)$$

$$\alpha = (1 - p_{\max})^{K-1}$$

$$x \geq y \iff {}^{\forall}k \in [|x|],\ x_k \geq y_k$$

$$x|_A = \{x_k |\ k \in A\}$$

$$x \odot y = [x_1 y_1, \ldots, x_{|x|} y_{|y|}]$$

$$V(x) = 1 - \prod_{k=1}^{K} (1 - x_k)$$

補題1

$${}^{\forall}x, y \in [0,1]^K \text{ s.t. } x \geq y,$$

$$V(x) - V(y) \leq \sum_{k=1}^{K} x_k - \sum_{k=1}^{K} y_k$$

補題2

$${}^{\forall}x, y \in [0, p_{\max}]^K \text{ s.t. } x \geq y,$$

$$\alpha \left[\sum_{k=1}^{K} x_k - \sum_{k=1}^{K} y_k \right] \leq V(x) - V(y)$$

補題の証明は簡単なので略

定理1：期待リグレットの上限（特殊形）

$\bar{v}(1) = \ldots = \bar{v}(K) = \gamma$ のとき $O\Big(\gamma(L-K)\log(n)\Big)$

$$R(n) \le \frac{\gamma}{\alpha} \sum_{e=K+1}^{L} \frac{(1+\varepsilon)\Delta_{e,K}(1+\log(1/\Delta_{e,K}))}{D_{\mathrm{KL}}(\bar{w}(e) \,\|\, \bar{w}(K))} \times (\log n + 3\log\log n) + C\,,$$

$$\textit{where } C = \tfrac{\gamma}{\alpha}\left(KL\tfrac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7K\log\log n\right)$$

証明：Cascade型の期待リグレット評価の形にうまく持っていく

$$\mathbf{R}_t = f(A^*, \mathbf{w}_t, \mathbf{v}_t) - f(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)$$

$$\mathcal{H}_t = (\mathbf{A}_1, \mathbf{c}_1, \ldots, \mathbf{A}_{t-1}, \mathbf{c}_{t-1}, \mathbf{A}_t)$$

とすると,

$$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] &= f(A^*, \bar{w}, \bar{v}) - f(\mathbf{A}_t, \bar{w}, \bar{v}) \\ &= V(\bar{w}|_{A^*} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}_t} \odot \bar{v})\end{aligned}$$ **関数Vの定義**

この期待値がかかるのは $\mathbf{w}_t$ と $\mathbf{v}_t$ のみなので計算可能

定理1：証明のつづき

- $\bar{v}(1) = \ldots = \bar{v}(K) = \gamma$ なのでA*の順番をどう並び替えても最適

$A^*(k) = A_t(k) \text{ if } A_t(k) \in A^*$ になるようA*を並び替えると,

$\bar{w}|_{A^*} \odot \bar{v} \geq \bar{w}|_{\mathbf{A}_t} \odot \bar{v}$ を満たすので,

$$\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] = V(\bar{w}|_{A^*} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}_t} \odot \bar{v})$$

補題1 $\longrightarrow$ $$\leq \gamma \left[\sum_{k=1}^{K} \bar{w}(a_k^*) - \sum_{k=1}^{K} \bar{w}(\mathbf{a}_k^t)\right]$$

補題2 $\longrightarrow$ $$\leq \frac{\gamma}{\alpha}\left[\underline{f_K(A^*, \bar{w}) - f_K(\mathbf{A}_t, \bar{w})}\right]$$

Cascade rewardの差が出てきた！

- 期待リグレットの上限は以下のように抑えられるので，命題1でバウンド

$$R(n) = \textstyle\sum_{t=1}^{n} \mathbb{E}\left[\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right]\right]$$

$$\leq \frac{\gamma}{\alpha} \sum_{t=1}^{n} \mathbb{E}\left[f_K(A^*, \bar{w}) - f_K(\mathbf{A}_t, \bar{w})\right] = \frac{\gamma}{\alpha} R_K(n) \quad \blacksquare$$

準備：一般形の上限を示すための補題

- 配列xを降順で並び替えた配列をx'とする

補題3

$$^\forall x \in [0,1]^K,\ ^\forall c \in [0,1]^K \text{ s.t. } c_1 \geq \ldots c_K,$$

$$V(c \odot x') - V(c \odot x) \leq \sum_{k=1}^{K} c_k x'_k - \sum_{k=1}^{K} c_k x_k$$

定理2：期待リグレットの上限（一般形）

gはKの増加正値関数？

$\bar{v}(1) \geq \ldots \geq \bar{v}(K)$ のとき $O\Big(\gamma(L-g(K))\log(n)\Big)$

$$R(n) \leq (1+\varepsilon)\sum_{i=1}^{K}\frac{\bar{v}(i)-\bar{v}(i+1)}{\alpha} \times \sum_{e=i+1}^{L}\frac{\Delta_{e,i}(1+\log(1/\Delta_{e,i}))}{D_{\mathrm{KL}}(\bar{w}(e)\,\|\,\bar{w}(i))}(\log n + 3\log\log n) + C\,,$$

$$\textit{where } \bar{v}(K+1)=0,\;\; C=\textstyle\sum_{i=1}^{K}\frac{\bar{v}(i)-\bar{v}(i+1)}{\alpha}\Big(iL\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}}+7i\log\log n\Big)$$

証明：Cascade型の期待リグレット評価の形にうまく持っていく

$$\mathbf{R}_t = f(A^*, \mathbf{w}_t, \mathbf{v}_t) - f(\mathbf{A}_t, \mathbf{w}_t, \mathbf{v}_t)$$

$$\mathcal{H}_t = (\mathbf{A}_1, \mathbf{c}_1, \ldots, \mathbf{A}_{t-1}, \mathbf{c}_{t-1}, \mathbf{A}_t)$$

とすると,

$$\mathbb{E}\left[\mathbf{R}_t \mid \mathcal{H}_t\right] = [V(\bar{w}|_{A^*}\odot\bar{v}) - V(\bar{w}|_{\mathbf{A}'_t}\odot\bar{v})] + [V(\bar{w}|_{\mathbf{A}'_t}\odot\bar{v}) - V(\bar{w}|_{\mathbf{A}_t}\odot\bar{v})]$$

（ここで，Atをwの降順に並べかえた配列をAt'とした）

定理2：証明のつづき

- **At'を導入した理由：** $\bar{w}|_{A^*} \odot \bar{v} \geq \bar{w}|_{\mathbf{A}_t} \odot \bar{v}$ が保証できないので，
定理1で使用した補題1と補題2の前提条件を満たせない

➡ $\bar{w}|_{A^*} \odot \bar{v} \geq \bar{w}|_{\mathbf{A}'_t} \odot \bar{v}$ **は満たされるので,**

補題1 ⟶ $$[V(\bar{w}|_{A^*} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}'_t} \odot \bar{v})] \leq \sum_{k=1}^{K} \bar{v}(k)(\bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^{t\,\prime}))$$

補題3 ⟶ $$[V(\bar{w}|_{\mathbf{A}'_t} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}_t} \odot \bar{v})] \leq \sum_{k=1}^{K} \bar{v}(k)(\bar{w}(\mathbf{a}_k^{t\,\prime}) - \bar{w}(\mathbf{a}_k^t))$$

➡ $$\begin{aligned}\mathbb{E}\left[\mathbf{R}_t \,|\, \mathcal{H}_t\right] &= [V(\bar{w}|_{A^*} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}'_t} \odot \bar{v})] + [V(\bar{w}|_{\mathbf{A}'_t} \odot \bar{v}) - V(\bar{w}|_{\mathbf{A}_t} \odot \bar{v})] \\ &\leq \sum_{k=1}^{K} \bar{v}(k)(\bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^t)) \\ &= \sum_{i=1}^{K} [\bar{v}(i) - \bar{v}(i+1)] \boxed{\sum_{k=1}^{i} (\bar{w}(a_k^*) - \bar{w}(\mathbf{a}_k^t))},\end{aligned}$$

where $\bar{v}(K+1) = 0$

補題2を適用 $\boxed{\leq \dfrac{R_i(n)}{\alpha}}$ ■

準備：B_{LB} 問題

定義 [DCMにおける B_{LB} 問題]

想定するパラメータが以下の仮定を満たす問題を B_{LB} 問題と呼ぶ

$$\bar{w}(e) = \begin{cases} p & e \in A^* \\ p - \Delta & \text{otherwise} \end{cases}$$

$$\bar{v}(1) = \ldots = \bar{v}(K) = \gamma$$

$B_{\mathrm{LB}}(L, A^*, p, \Delta, \gamma)$ ←のように表記

- **最適なアイテム列に含まれる各アイテムに魅了される確率は p で共通**
- **最適なアイテム列に含まれない各アイテムに魅了される確率は p - Δ で共通**

準備：下限を示すための補題

補題4

$^\forall x, y \in [0,1]^K$ s.t. $x \geq y, \quad \gamma \in [0,1]$.

$$V(\gamma x) - V(\gamma y) \geq \gamma[V(x) - V(y)].$$

定理3：期待リグレットの下限（特殊形）

$\bar{v}(1) = \ldots = \bar{v}(K) = \gamma$ のとき $\Omega\left(\gamma(L-K)\frac{\Delta}{D_{\mathrm{KL}}(p-\Delta \,\|\, p)}\log n\right)$

$$\boxed{\liminf_{n\to\infty} \frac{R(n)}{\log n} \geq \gamma\alpha \frac{(L-K)\Delta}{D_{\mathrm{KL}}(p-\Delta \,\|\, p)} \quad {}^{\forall}\textit{DCM bandit } B_{\mathrm{LB}}}$$

証明：各アイテムをDBに入れた回数 $\mathbf{T}_n(e)$ で評価する形に持っていく

補題4 ⟶ $$R(n) \geq \gamma \mathbb{E}\left[\sum_{t=1}^{n}(f_K(A^*, \mathbf{w}_t) - f_K(\mathbf{A}_t, \mathbf{w}_t))\right]$$

補題2 ⟶ $$\geq \gamma\alpha\mathbb{E}\left[\sum_{t=1}^{n}\left(\sum_{k=1}^{K}\mathbf{w}_t(a_k^*) - \sum_{k=1}^{K}\mathbf{w}_t(\mathbf{a}_k^t)\right)\right]$$

B_{LB}の仮定 ⟶ $$\geq \gamma\alpha\Delta \sum_{e=K+1}^{L} \mathbb{E}\left[\underline{\mathbf{T}_n(e)}\right]$$

各アイテムについて，実際にアイテムとして出された回数 ≥ DBに入った回数

定理3：証明のつづき

Lai & Robbins (1985)

Fact：強一致性を満たすアルゴリズムの$\mathbf{T}_n(e)$の下限

任意の強一致性を満たすアルゴリズムに対して以下が成立

$$\liminf_{n\to\infty} \frac{\mathbb{E}\left[\mathbf{T}_n(e)\right]}{\log n} \geq \frac{\Delta}{D_{\mathrm{KL}}(p-\Delta \,\|\, p)}$$

よって素直に計算すると…

$$\liminf_{n\to\infty} \frac{R(n)}{\log n} \geq \liminf_{n\to\infty} \gamma\alpha\Delta \sum_{e=K+1}^{L} \frac{\mathbb{E}[\mathbf{T}_n(e)]}{\log(n)}$$

$$\geq \liminf_{n\to\infty} \gamma\alpha(L-K)\Delta \frac{\Delta}{D_{\mathrm{KL}}(p-\Delta)||p} \quad \blacksquare$$

※論文のregretの下限をよく見ると…↑より Δが一つ少ない…

- これ，Factの式が間違ってる（分子にΔは要らない）のでは…?? [要確認]

問題設定

- 研究背景
- 問題設定
- 提案手法
- 理論解析
- 実験結果
- まとめ，今後の流れ

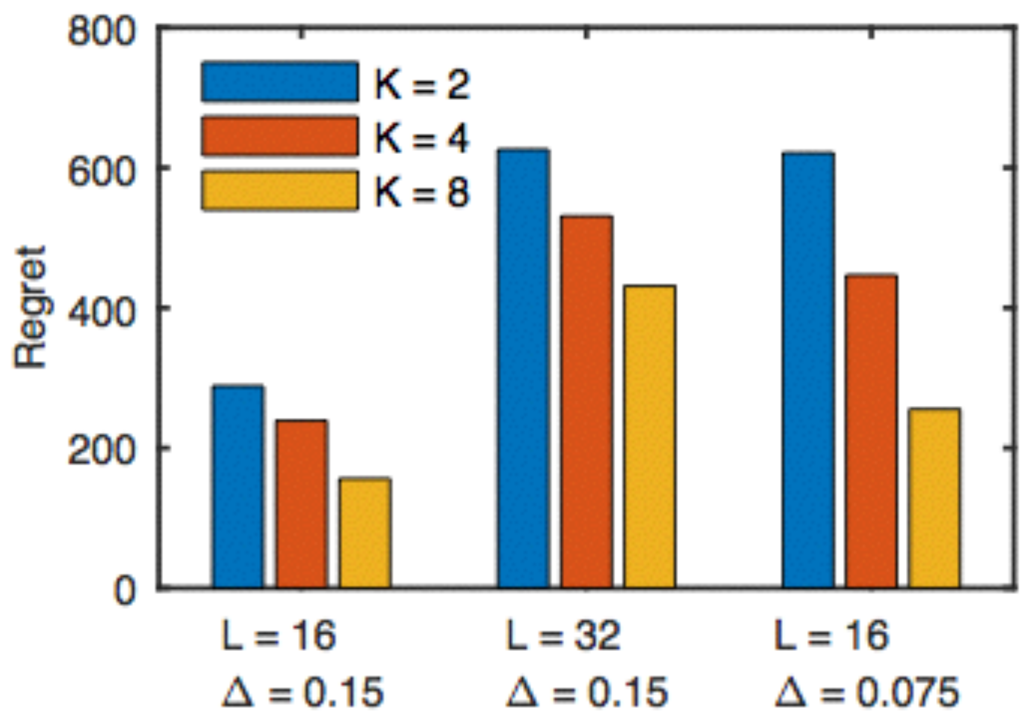
実験1：人工データによるRegret上限の検証

定理1で示したバウンドの正しさをある程度確認できた

- 以下の B_{LB} 問題を生成し，パラメータを変えてRegret計算

$$B_{\mathrm{LB}}(L, A^* = [K], p = 0.2, \Delta, \gamma)$$

Figure 2a　$\gamma = 0.8$ と固定，L, K, Δ を変化



上限：$O\left(\gamma(L-K)\frac{\Delta(1+\log(1/\Delta))}{D_{\mathrm{KL}}(p-\Delta \,\|\, p)}\log n\right)$

- 上限の $O(L-K)$ 部分が反映
- Δとは逆相関　←KL由来？

ステップ数は n = 10^5で固定

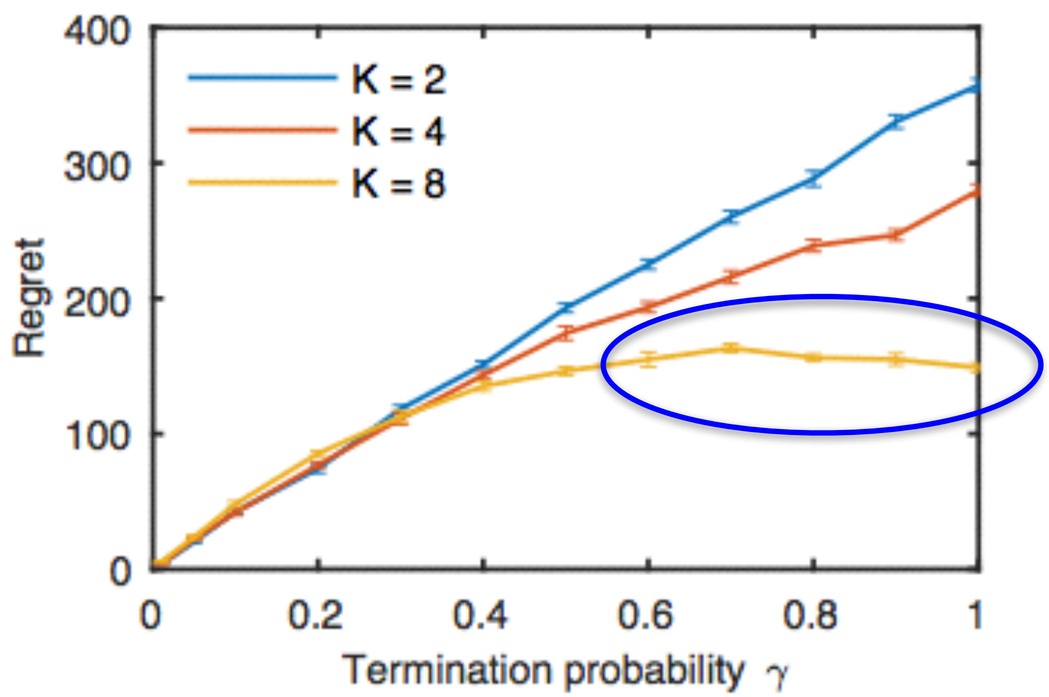
実験1：人工データによるRegret上限の検証

定理1で示したバウンドの正しさをある程度確認できた

- 以下の B_{LB} 問題を生成し，パラメータを変えてRegret計算

$$B_{\mathrm{LB}}(L, A^* = [K], p = 0.2, \Delta, \gamma)$$

Figure 2b $L = 10, \Delta = 0.15$ と固定，K, γ を変化



上限：$O\left(\gamma(L-K)\frac{\Delta(1+\log(1/\Delta))}{D_{\mathrm{KL}}(p-\Delta \,\|\, p)}\log n\right)$

- $p < 1/K \Rightarrow O(\gamma)$ 部分が反映
- p, K が大きいとき上限は緩そう

ステップ数は n = 10^5で固定

実験2：提案手法と他の手法との比較

データの入れ方が異なる2手法と，アルゴリズムが異なる1手法

$\mathbf{A}_t = (1,2,3,4)$, $\mathbf{c}_t = (0,1,1,0)$ となる場合を例にして解説

- **First Click：ctが最初のクリックのみを含む**DCM Bandits
  - DBに保存する（アイテム, クリック）列は　(1,0), (2,1)？
- **First Click：ctが最後のクリックのみを含む**DCM Bandits
  - DBに保存する（アイテム, クリック）列は　(1,0), (2,0), (3,1)？
- **Ranked KL-UCB：各場所に対し独立に普通のKL-UCB**を適用
  - DBに保存する（場所, アイテム, クリック）列は
    (1,1,0), (2,2,1), (3,3,1), (4,4,0)？

- **提案手法：ctが全てのクリック情報を含む**DCM Bandits
  - DBに保存する（アイテム, クリック）列は　(1,0), (2,1), (3,1)

実験2：人工データによる他の手法との比較

提案したdcmKL-UCBが最小のRegretを達成！

- 以下の B_{LB} 問題を生成し，手法を変えて各ステップでRegret計算

$$B_{\mathrm{LB}}(L=16, A^*=[4], p=0.2, \Delta=0.15, \gamma=0.5)$$

Figure 3a Cascade的な手法群との比較

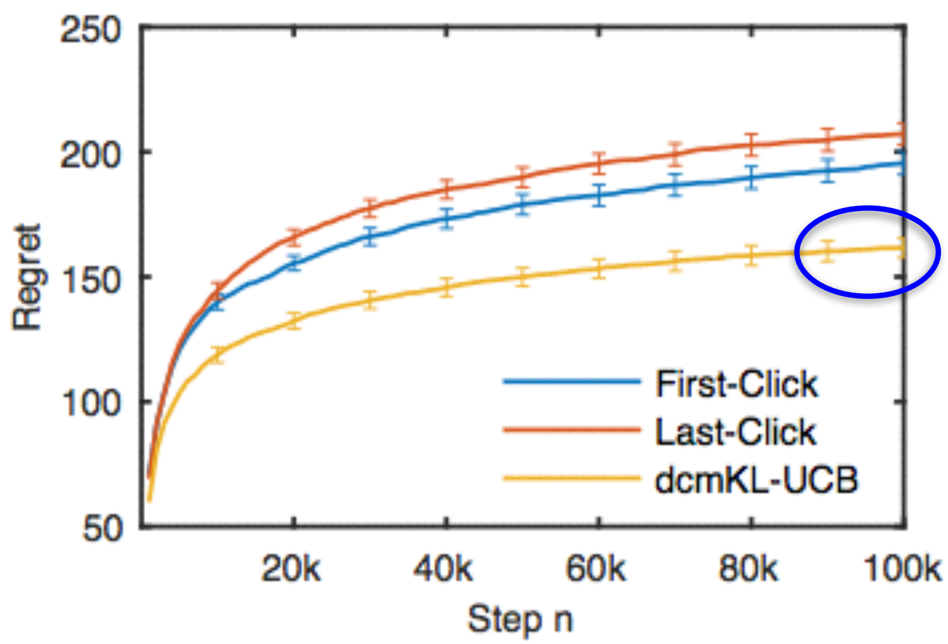
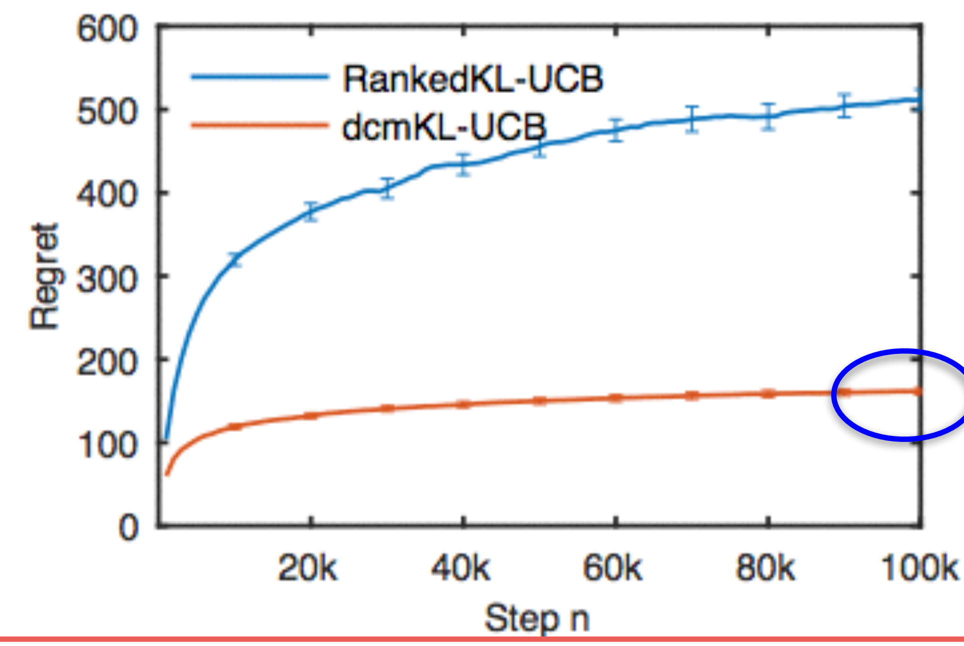


Figure 3b RankedKL-UCBとの比較



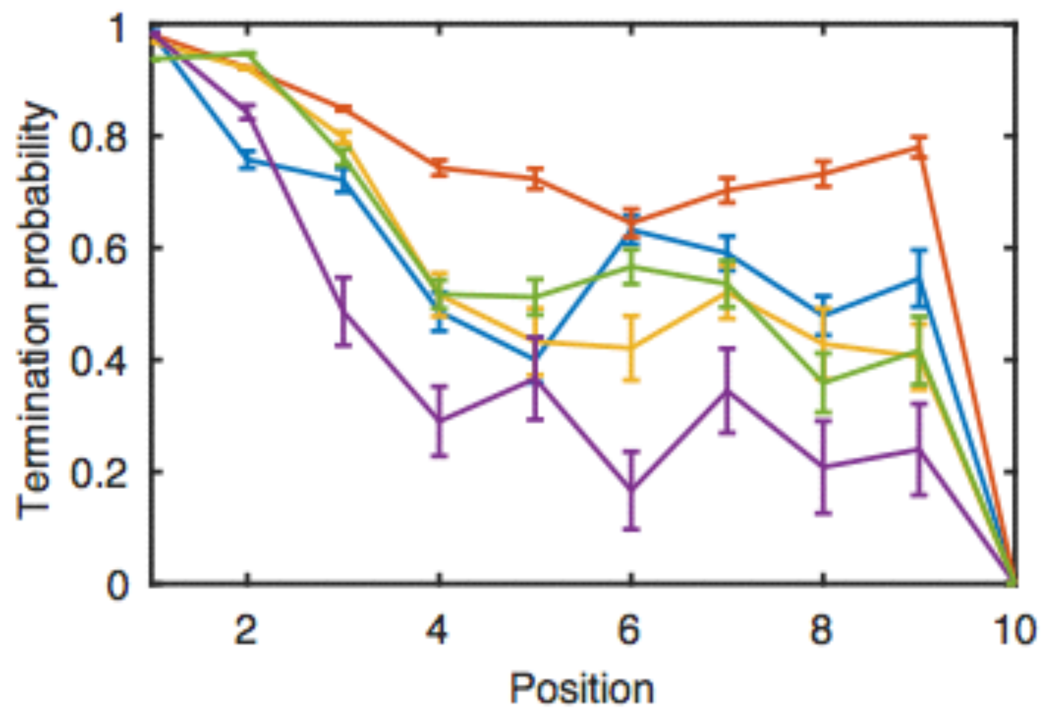
- **提案手法は概ね $O(\log(n))$ を達成**
- **特に, $O(KL)$ なRankedKL-UCBとは “K=4” 倍の差があり, dcmKL-UCBが $O(L-K)$ であることと整合的**

実験3：実データの特徴

$\bar{v}(1) \geq \ldots \geq \bar{v}(K)$ という仮定は満たされていない

- Yandex datasetから頻出トップ5クエリのモデルを，DCMで事前に教師あり学習し，場所ごとのクリック後満足確率を比較

Figure 2c Termination確率の比較（ $K = 10$ ）



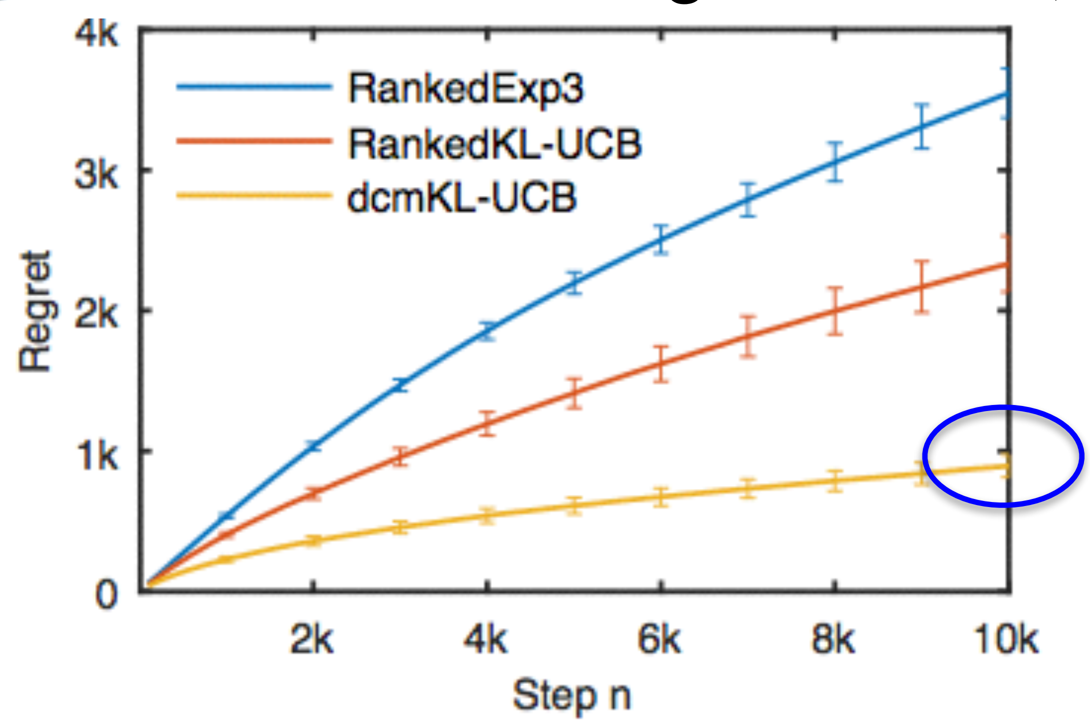
場所1はクリック後ほぼ確実に満足するが，それ以降は降順でない

実験3：実データによる他の手法との比較

提案したdcmKL-UCBが最小のRegretを達成！

- Yandex datasetから頻出トップ20クエリのモデルを，DCMで事前に教師あり学習し，それに基づきデータを生成

Figure 3c 手法ごとのRegretの挙動（ $K = 10$ ）



※Lは未記入

- 提案手法は概ね $O(\log(n))$ を達成
- Termination確率の順序についての仮定が満たされていない実データでも，うまくいっている！

問題設定

- 研究背景
- 問題設定
- 提案手法
- 理論解析
- 実験結果
- まとめ，今後の流れ

まとめ【概要再掲】

- **背景：検索システムにおけるユーザの満足度を最大化したい**
  - ユーザの行動モデルについては様々なものが提案されているが，そのほとんどは学習データがすでに揃っていることが前提
  - 適当な行動モデルを仮定したときに，学習データを集めつつユーザの満足度を最大化する方策を考えたい

- **問題設定：Dependent Click Modelの下でのBandit問題**
  - Cascade Modelの下でのBandit問題 [Kveton et al., (2015a)] の拡張

- **提案手法：dcmKL-UCB（KL-UCBの拡張）**
  - 適当な仮定の下で期待リグレットの上限と下限を導出
  - 仮定が満たされるか否かに関わらず，実験性能が良いことを確認
    - ✳ 複数クリックを扱うモデルでは初のregret最適保証つき逐次学習手法

今後の流れ（どれかやってみたい）

- **実装**（気が向いたら…多分そんなに難しくない）
- **DCM UCB1を提案**
  - 2015年にCascade KL-UCBと同時にCascade UCB1が提案済み
- **DCM BanditsをContextualな問題設定に拡張**
  - [Zong et al. (2016)] ではCascade LinUCBが提案済み
  - DCMUCB1が提案できれば**DCM LinUCB**ができそう？
- **別のユーザ行動モデルを仮定した逐次学習手法を提案**
  - UBM, CCM, DBNなどDCMとほぼ同時期に提案されたモデルは多数
  - ICML 2017には**UBM Bandits, CCM Bandits**が出てきそう？
- **バウンド改善**（自分では多分ムリ…）

疑問点などメモ

- **証明についての疑問**
  - 命題1はもう少し厳密に示さなくてよいのか？
  - 補題1, 2で関数d(x)が正であると示しているが，これは不要では？
  - 補題3の “Then” 直後の x_k’ は \tilde{x}_k では？
  - 定理3でΔを一つ掛け忘れているのでは？
- **定理2で“gはKの増加正値関数”と解釈したが，これでよいか**
- **Algorithm内で実際にfの値を計算する必要はあるのか**
  - UCBの順序とvの順序を合わせるだけでよいのでは

各種参考文献（ユーザの行動モデル）

Cascade model

Craswell, Nick, Zoeter, Onno, Taylor, Michael, and Ram- sey, Bill. An experimental comparison of click position- bias models. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining*, pp. 87–94, 2008.

User Behavior model (UBM)

G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. *In SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 331–338, 2008.

Click Chain model (CCM)

Guo, Fan, Liu, Chao, Kannan, Anitha, Minka, Tom, Taylor, Michael, Wang, Yi Min, and Faloutsos, Christos. Click chain model in web search. In *Proceedings of the 18th International Conference on World Wide Web*, pp. 11– 20, 2009a.

Dependent Click model (DCM)：Notationが少しわかりにくい論文なので注意

Guo, Fan, Liu, Chao, and Wang, Yi Min. Efficient multiple-click models in web search. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pp. 124–131, 2009b.

Dynamic Bayesian Network (DBN)

Chapelle, Olivier and Zhang, Ya. A dynamic bayesian network click model for web search ranking. In *Proceed- ings of the 18th International Conference on World Wide Web*, pp. 1–10, 2009.

各種参考文献（Bandit関係）

KL-UCB

Chapelle, Olivier and Zhang, Ya. A dynamic bayesian net- work click model for web search ranking. In *Proceedings of the 18th International Conference on World Wide Web*, pp. 1–10, 2009.

RankedKL-UCB

Slivkins, Aleksandrs, Radlinski, Filip, and Gollapudi, Sreenivas. Ranked bandits in metric spaces: Learning di- verse rankings over large document collections. *Journal of Machine Learning Research*, 14(1):399–436, 2013.

Cascading Bandits (based on UCB1 and KL-UCB)

Kveton, Branislav, Szepesvari, Csaba, Wen, Zheng, and Ashkan, Azin. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015a.

Multiple Play Bandit (based on Thompson Sampling)

J. Komiyama, J. Honda, and H. Nakagawa. Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. *In Proceedings of the 32nd International Conference on Machine Learning*, pp 1152–1161, 2015.

Cascading Contextual Bandits (based on LinUCB)

Zong, Shi, Ni, Hao, Sung, Kenny, Ke, Nan Rosemary, Wen, Zheng, and Kveton, Branislav. Cascading bandits for large-scale recommendation problems. In *Proceedings of the 32nd Conference on Uncertainty in Artificial In-telligence*, 2016.

Appendix

Cascading Banditsとの比較

DCM Banditsとの違いは，データの入れ方（とRegretの測り方）

ここは一見違うが，\bar{v}を降順に並べると同じ

Algorithm 1 UCB-like algorithm for cascading bandits.

// Initialization
Observe $\mathbf{w}_0 \sim P$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for all $t = 1, \dots, n$ **do**
  Compute UCBs $\mathbf{U}_t(e)$ (Section 3.2)

  // Recommend a list of K items and get feedback
  Let $\mathbf{a}_1^t, \dots, \mathbf{a}_K^t$ be K items with largest UCBs
  $\mathbf{A}_t \leftarrow (\mathbf{a}_1^t, \dots, \mathbf{a}_K^t)$
  Observe click $\mathbf{C}_t \in \{1, \dots, K, \infty\}$

  // Update statistics
  $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
  for all $k = 1, \dots, \min\{\mathbf{C}_t, K\}$ **do**
    $e \leftarrow \mathbf{a}_k^t$
    $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
    $\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) \leftarrow \dfrac{\mathbf{T}_{t-1}(e)\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbb{1}\{\mathbf{C}_t = k\}}{\mathbf{T}_t(e)}$

Cascading Bandits (Kveton et al,)

Algorithm 1 dcmKL-UCB for solving DCM bandits.

// Initialization
Observe $\mathbf{w}_0 \sim P_{\mathrm{w}}$
$\forall e \in E : \mathbf{T}_0(e) \leftarrow 1$
$\forall e \in E : \hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e)$

for all $t = 1, \dots, n$ **do**
  for all $e = 1, \dots, L$ **do**
    Compute UCB $\mathbf{U}_t(e)$ using (1)

  // Recommend and observe
  $\mathbf{A}_t \leftarrow \arg\max_{A \in \Pi_K(E)} f(A, \mathbf{U}_t, \tilde{v})$
  Recommend $\mathbf{A}_t$ and observe clicks $\mathbf{c}_t \in \{0, 1\}^K$
  $\mathbf{C}_t^{\mathrm{last}} \leftarrow \max\{k \in [K] : \mathbf{c}_t(k) = 1\}$

  // Update statistics
  $\forall e \in E : \mathbf{T}_t(e) \leftarrow \mathbf{T}_{t-1}(e)$
  for all $k = 1, \dots, \min\{\mathbf{C}_t^{\mathrm{last}}, K\}$ **do**
    $e \leftarrow \mathbf{a}_k^t$
    $\mathbf{T}_t(e) \leftarrow \mathbf{T}_t(e) + 1$
    $\hat{\mathbf{w}}_{\mathbf{T}_t(e)}(e) \leftarrow \dfrac{\mathbf{T}_{t-1}(e)\hat{\mathbf{w}}_{\mathbf{T}_{t-1}(e)}(e) + \mathbf{c}_t(k)}{\mathbf{T}_t(e)}$

Dcm Bandits (紹介論文)

- RegretをCascade型で統一すると，複数クリックがないとき同じ挙動
- データが同じとき，学習に使われるデータはDCM Banditsの方が多い or 同数

Cascade KL-UCBの証明

- アイテム $e^* \in A^*$ と $e \notin A^*$ の魅力度のgapを以下のように表記

$$\Delta_{e,e^*} = \bar{w}(e^*) - \bar{w}(e)$$

- $\bar{w}(1) \geq \ldots \geq \bar{w}(L)$ と仮定しても一般性を失わない

Cascade KL-UCBアルゴリズムを用いてK個のアイテムを出すとき,
Cascade型の期待リグレットの上限は…

任意の $\epsilon > 0$ に対して，ある正値関数 $C_2(\epsilon)$, $\beta(\epsilon)$ が存在し,

$$R_K(n) \leq \sum_{e=K+1}^{L} \frac{(1+\varepsilon)\Delta_{e,K}(1+\log(1/\Delta_{e,K}))}{D_{\mathrm{KL}}(\bar{w}(e) \parallel \bar{w}(K))} \times (\log n + 3\log\log n) + C\,,$$

$$\textit{where } C = KL\frac{C_2(\varepsilon)}{n^{\beta(\varepsilon)}} + 7K\log\log n$$

Cascade KL-UCBの証明

- 各時刻で，少なくとも一つの最適なアイテムにおいて，
  真のattractive確率がそのアイテムのUCBを超える事象を定義

$$\mathcal{E}_t = \{\exists 1 \le e \le K \text{ s.t. } \bar{w}(e) > \mathbf{U}_t(e)\}$$

- 上記事象とその補集合を使ってCascade型リグレットを分解

$$R_K(n) = \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}\{\mathcal{E}_t\}\,\mathbf{R}_t\right] + \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}\{\overline{\mathcal{E}}_t\}\,\mathbf{R}_t\right]$$

- 第一項はKL-UCBの元論文の定理10にunion boundを適用

$$\mathbb{E}\left[\sum_{t=1}^{n} \mathbb{1}\{\mathcal{E}_t\}\,\mathbf{R}_t\right] \le 7K \log\log n$$

- 第二項は長いので略（KL-UCBの元論文の補題8を使用）