



人間行動モデリングと数理最適化

日本OR学会研究部会：
最適化の理論とアルゴリズム（RAOTA）第1回研究会

2023年6月17日（土）@ 国立情報学研究所

赤木 康紀（NTT 人間情報研究所）

自己紹介

- **名前:** 赤木 康紀 (あかぎ やすのり)



- **経歴:**

- ✓ 2014年4月 - 2016年3月:

東京大学大学院 情報理工学系研究科 数理情報学専攻 修士課程

- ✓ 2016年4月 - 現在:

NTT 研究所

- ✓ 2022年4月 - 2023年3月:

筑波大学大学院 理工情報生命学術院 システム情報工学研究群

情報理工学位プログラム 博士後期課程

- **研究:** 数理モデリング, 数理最適化, データマイニング など

- **趣味:** ビール, 料理 など

- ## — 人間情報研究所の技術カテゴリ

AR/VR、UI



2

- 様々なデバイスにより収集されデジタル化された人の行動データを計算論的アプローチによって扱う **Behavioral Data Science** という研究領域が盛り上がりを見せている
- この領域において、**数理的なモデリング** に基づいて**人間の行動を推定・予測・介入**する技術を研究開発

例:

- ✓ 人間の毎日の運動の継続の様子を数理的にモデリングし、メッセージングなどで介入することで運動の継続を支援
- ✓ 人間の地理空間上での移動の様子をモデリングし、混雑を避けるようにサイネージなどで行動変容

人間行動モデリングにおいて、**数理最適化技術は非常に重要**

- データが大量にある場合が多く
機械学習モデルで学習・推論を行う際 に
効率的に最適化問題を解く必要がある
- 学習したモデルに基づいて **最適な介入を求める際** には
何らかの最適化問題を解く必要がある

講演者が今まで行ってきた

人間行動モデリング × 数理最適化 に関する研究について紹介

- **前半:**

離散 DC アルゴリズムによる

Collective Graphical Model の近似を用いない MAP 推定

- **後半:**

解析的に扱いやすい

現在バイアスの影響下の人間行動のモデリング手法

※ 2つの研究内容は独立です

離散 DC アルゴリズムによる Collective Graphical Model の 近似を用いない MAP 推定

Y. Akagi, N. Marumo, H. Kim, T. Kurashima, H. Toda.

“Non-approximate Inference for Collective Graphical Models on Path Graphs
via Discrete Difference of Convex Algorithm”

NeurIPS'21

前半: 目次

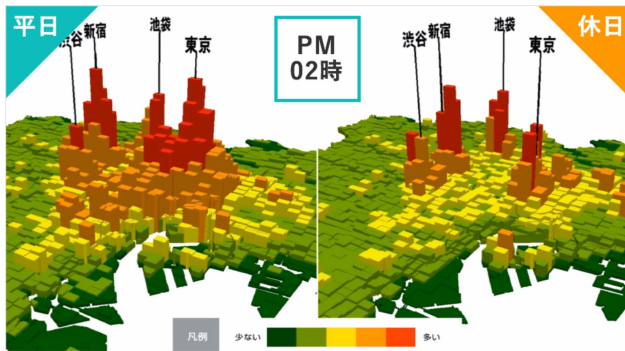
- **研究の背景**
- Collective Graphical Model (CGM)
- CGM における MAP 推定問題
- 提案手法
- 実験

集計データ

- 通常のデータマイニング・機械学習技術は
個人ごとのデータやラベルが手に入る状況 を想定している場合が多い
- しかし近年、個人のデータを手に入れることが難しく、
複数の個人のデータを集計したデータしか手に入らない場合がある
- このような **集計データの解析を可能にする技術** の重要性が増し、
様々な文脈で研究が行われるようになっている

集計された人の位置情報データ: 人流情報

- 特に、位置情報は個人のトラッキングの難しさ・プライバシーへの配慮の必要性から、集計されている状態でしか手に入らない場合も多い



モバイル空間統計

NTT ドコモの携帯電話ネットワークの情報を用いて作成される人口の統計情報

(https://mobaku.jp/service/jpn_distribution/)

- 集計された位置情報データを有効活用する方法論は、モビリティ分野やマーケティング分野への有効活用が期待され、社会的に非常に価値が大きい

扱いたいタスク ①

各時刻の人口分布から各時刻間の移動人数を推定

入力: 各時刻・各エリアの人口

時刻 1

10	20
20	30

時刻 2

20	10
20	30

時刻 3

0	10
40	30

出力: 各時刻間における各エリア間の移動人数

時刻 1 と 2 の間

←	
10 人	

時刻 2 と 3 の間

↓	
20 人	

活用先: 群衆の移動の向きを考慮した移動サービス・マーケティング

扱いたいタスク ②

2つの時刻の人口からその間の人口を補間

入力: 時刻 1・4 における人口

時刻 1

8	0	0
0	0	0
0	0	0

時刻 2

?

時刻 3

?

時刻 4

0	0	0
0	0	0
4	0	4

出力: 時刻 2・3 における人口

時刻 2

2	0	0
3	3	0
0	0	0

時刻 3

0	0	0
2	2	0
2	0	2

人口データが荒い時間粒度でしか入手できない場合に有用

集計された位置情報データを扱う手法

- 回帰に基づく手法

- ✓ 入力 x と出力 y の間の関係を
なんらかの関数でモデリングすることで推定や予測を行う
- ✓ **学習のために教師データが必要**

(代表例)

ガウシアンマルコフ確率場 + 空間的残差モデル [Hoang et al., 2016]

Deep Spatio-Temporal Residual Network [Zhang et al., 2017]

- **確率的生成モデルに基づく手法**

- ✓ データの生成過程を確率的にモデリングし,
観測が与えられた下での事後分布に基づいて推定を行う
- ✓ **教師なし**の設定でも使用可能

(代表例)

Collective Graphical Model [Sheldon and Dietterich, 2011]

前半: 目次

- 研究の背景
- **Collective Graphical Model (CGM)**
- CGM における MAP 推定問題
- 提案手法
- 実験

Collective Graphical Model (CGM)

集計された位置情報を扱うための確率的生成モデル

個人の移動モデル（グラフィカルモデル）

時刻 1 時刻 2 時刻 3



ϕ_{1ij} ϕ_{2ij}

X_t : 時刻 t にどの
エリアにいるか?
を表す確率変数

ϕ_{tij} : 時刻 $t \rightarrow t+1$
で移動 $i \rightarrow j$ の起こりやすさ
(グラフィカルモデルの
ポテンシャル関数)

Collective Graphical Model (CGM)

集計された位置情報を扱うための確率的生成モデル

個人の移動モデル

時刻 1 時刻 2 時刻 3

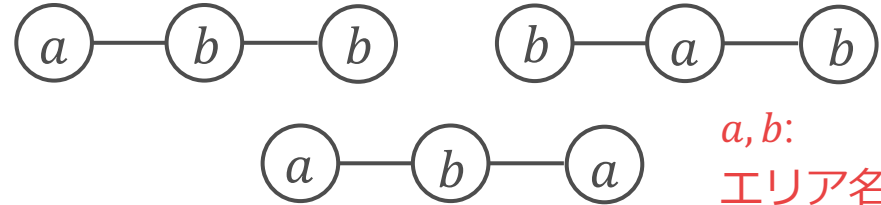


X_t : 時刻 t にどの
エリアにいるか?
を表す確率変数

サンプリング



個人ごとの実際の移動パターン



a, b :
エリア名

Collective Graphical Model (CGM)

集計された位置情報を扱うための確率的生成モデル

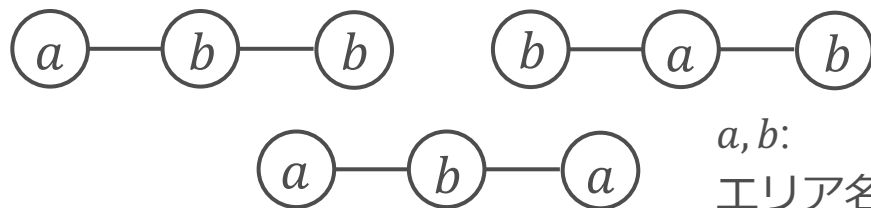
個人の移動モデル

時刻 1 時刻 2 時刻 3



X_t : 時刻 t にどの
エリアにいるか?
を表す確率変数

個人ごとの実際の移動パターン



真の人口・移動人数

時刻 1 時刻 2 時刻 3

エリア	人口
a	2
b	1

...

...

n_{ti} :
時刻 t
エリア i
の真の人口

時刻 1 → 2 時刻 2 → 3

	a	b
a	0	2
b	1	0

...

n_{tij} :
時刻 $t \rightarrow t+1$
エリア $i \rightarrow j$
移動した真の人数

集計

Collective Graphical Model (CGM)

集計された位置情報を扱うための確率的生成モデル

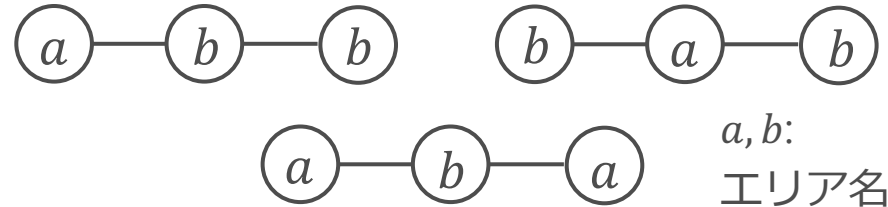
個人の移動モデル

時刻 1 時刻 2 時刻 3



X_t : 時刻 t にどの
エリアにいるか?
を表す確率変数

個人ごとの実際の移動パターン



真の人口・移動人数

時刻 1 時刻 2 時刻 3

エリア	人口
a	2
b	1

... ...

n_{ti} : 観測ノイズ
時刻 t を加える
エリア i
の真の人口

観測される人口・移動人数

時刻 1 時刻 2 時刻 3

エリア	人口
a	3
b	1

... ...

y_{ti} :
時刻 t
エリア i
の観測人口

時刻 1 → 2 時刻 2 → 3

	a	b
a	0	2
b	1	0

...

n_{tij} :
時刻 $t \rightarrow t+1$
エリア $i \rightarrow j$
移動した真の人数

時刻 1 → 2 時刻 2 → 3

	a	b
a	1	2
b	2	0

...

y_{tij} :
時刻 $t \rightarrow t+1$
エリア $i \rightarrow j$
移動した
観測人数

CGMによるタスクの解き方の例 ①

各時刻の人口分布から各時刻間の移動人数を推定するタスク

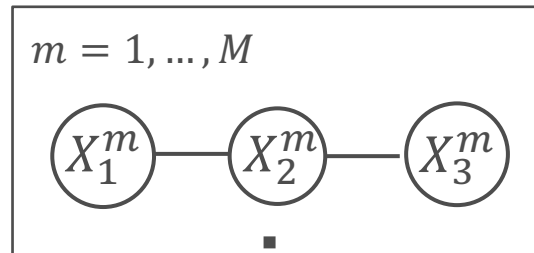
入力: 各時刻・各エリアの人口

時刻 1		時刻 2		時刻 3	
10	20	20	10	0	10
20	30	20	30	40	30

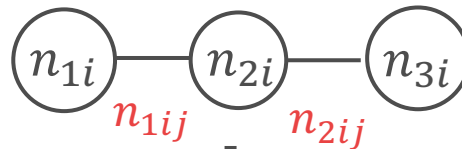
出力: 各時刻間における各エリア間の移動人数

時刻 1 と 2 の間		時刻 2 と 3 の間	
	←	↓	20 人

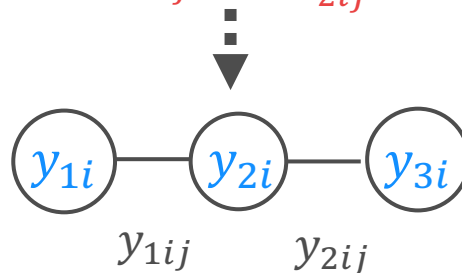
個人モデル



真の
人口・移動人数



観測される
人口・移動人数



入力

観測される人口 y_{1i}, y_{2i}, y_{3i}

出力

真の移動人数 n_{1ij}, n_{2ij}

CGMによるタスクの解き方の例 ②

人口分布を補間するタスク

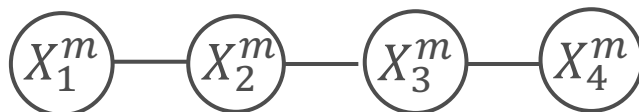
入力: 時刻 1・4 における人口

時刻 1			時刻 2	時刻 3	時刻 4		
8	0	0	?	?	0	0	0
0	0	0			0	0	0
0	0	0			4	0	4

出力: 時刻 2・3 における人口

時刻 2			時刻 3		
2	0	0	0	0	0
3	3	0	2	2	0
0	0	0	2	0	2

$m = 1, \dots, M$



入力

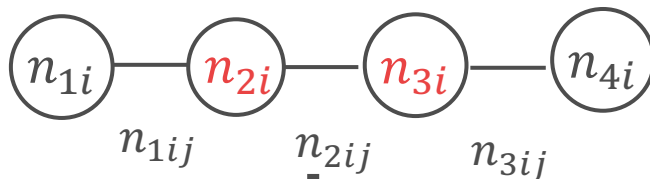
観測される人口 y_{1i}, y_{4i}

出力

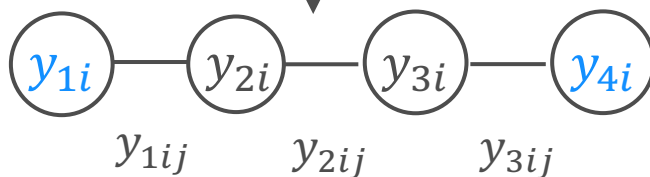
真の人口 n_{2i}, n_{3i}

個人モデル

真の
人口・移動人数



観測される
人口・移動人数



前半: 目次

- 研究の背景
- Collective Graphical Model (CGM)
- **CGM における MAP 推定問題**
- 提案手法
- 実験

CGM における MAP 推定問題

入力:

観測される人口・移動人数 y

※ 今回は簡単のため

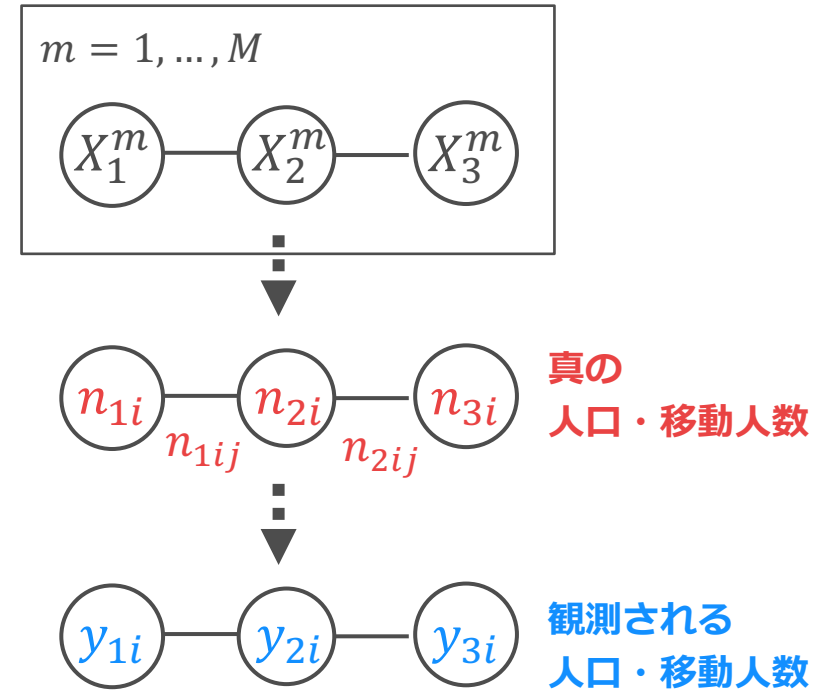
観測人口が全て与えられるとする

※ グラフィカルモデルのポテンシャルも
与えられるとする

出力:

事後確率 $\Pr(\mathbf{n}|\mathbf{y})$ を最大化する

真の人口・移動人数 \mathbf{n}



CGM の文脈において非常に重要視されている問題

- CGM における様々なタスク（移動人数の推定, 値の補間）
を解くために利用可能

MAP 推定問題の導出

- MAP 推定問題: $\max_{\mathbf{n}} \Pr(\mathbf{n}|\mathbf{y})$
- ベイズの定理より $\Pr(\mathbf{n}|\mathbf{y}) \propto \Pr(\mathbf{n}) \cdot \Pr(\mathbf{y}|\mathbf{n})$
- $\Pr(\mathbf{n})$: **CGM分布** 以下のように計算できる [Sun et al. 2015]

$$\Pr(\mathbf{n}) = \frac{M!}{Z^M} \cdot \frac{\prod_{t=2}^{T-1} \prod_{i \in \{1, \dots, R\}} n_{ti}!}{\prod_{t=1}^{T-1} \prod_{i \in \{1, \dots, R\}} \prod_{j \in \{1, \dots, R\}} n_{tij}!} \prod_{t=1}^{T-1} \prod_{i \in \{1, \dots, R\}} \prod_{j \in \{1, \dots, R\}} \phi_{tij}^{n_{tij}}$$

- $\Pr(\mathbf{y}|\mathbf{n})$: **観測分布** (ノイズ分布)

以下の仮定をおく

- ✓ 変数ごとの観測分布の **独立性**: $\Pr(\mathbf{y}|\mathbf{n}) = \prod_t \prod_i \Pr(y_{ti}|n_{ti})$
- ✓ 各 $\Pr(y_{ti}|n_{ti})$ が **対数凹分布**
(ガウス分布・ポアソン分布などメジャーな分布を含む)

- 目的関数の負の対数をとって最小化問題として定式化

MAP 推定問題の数学的表現

$$\begin{aligned}
 \min_{\mathbf{n}} \quad & \sum_{t=1}^{T-1} \sum_{i,j \in \{1, \dots, R\}} (\log n_{tij}! - n_{tij} \log \phi_{tij}) + \sum_{t=2}^{T-1} \sum_{i \in \{1, \dots, R\}} (-\log n_{ti}!) \\
 & + \sum_{t=1}^T \sum_{i \in \{1, \dots, R\}} (-\log p_{ti}(y_{ti}|n_{ti})) \\
 \text{s. t.} \quad & \sum_{i \in \{1, \dots, R\}} n_{ti} = M \quad t = 1, \dots, T \\
 & \sum_{j \in \{1, \dots, R\}} n_{tij} = n_{ti} \quad t = 1, \dots, T-1, i = 1, \dots, R \\
 & \sum_{j \in \{1, \dots, R\}} n_{tji} = n_{t+1,i} \quad t = 1, \dots, T-1, i = 1, \dots, R \\
 & n_{ti}, n_{tij} \in \mathbb{Z}_{\geq 0}
 \end{aligned}$$

- M : サンプルサイズ, T : 時刻数
- R : エリア数
- ϕ_{tij} : グラフィカルモデルのポテンシャル (辺 $(t, t+1)$, 値の組 (i, j))
- p_{ti} : 頂点 t における値 i の観測分布 (対数凹分布)

MAP 推定問題の数学的表現

$$\begin{aligned}
 \min_n \quad & \underbrace{\sum_{t=1}^{T-1} \sum_{i,j \in \{1, \dots, R\}} (\log n_{tij}! - n_{tij} \log \phi_{tij})}_{\text{真の移動人数に関する項}} + \underbrace{\sum_{t=2}^{T-1} \sum_{i \in \{1, \dots, R\}} (-\log n_{ti}!)}_{\text{真の人口に関する項}} \\
 & + \underbrace{\sum_{t=1}^T \sum_{i \in \{1, \dots, R\}} (-\log p_{ti}(y_{ti} | n_{ti}))}_{\text{観測ノイズに関する項}} \\
 \text{s. t.} \quad & \sum_{i \in \{1, \dots, R\}} n_{ti} = M \quad t = 1, \dots, T \\
 & \sum_{j \in \{1, \dots, R\}} n_{tij} = n_{ti} \quad t = 1, \dots, T-1, i = 1, \dots, R \\
 & \sum_{j \in \{1, \dots, R\}} n_{tji} = n_{t+1,i} \quad t = 1, \dots, T-1, i = 1, \dots, R \\
 & n_{ti}, n_{tij} \in \mathbb{Z}_{\geq 0}
 \end{aligned}$$

- M : サンプルサイズ, T : 時刻数
- R : エリア数
- ϕ_{tij} : グラフィカルモデルのポテンシャル (辺 $(t, t+1)$, 値の組 (i, j))
- p_{ti} : 頂点 t における値 i の観測分布 (対数凹分布)

MAP 推定問題の数学的表現

$$\begin{aligned}
 \min_n \quad & \sum_{t=1}^{T-1} \sum_{i,j \in \{1, \dots, R\}} (\log n_{tij}! - n_{tij} \log \phi_{tij}) + \sum_{t=2}^{T-1} \sum_{i \in \{1, \dots, R\}} (-\log n_{ti}!) \\
 & + \sum_{t=1}^T \sum_{i \in \{1, \dots, R\}} (-\log p_{ti}(y_{ti} | n_{ti})) \\
 \text{s.t.} \quad & \sum_{i \in \{1, \dots, R\}} n_{ti} = M \quad t = 1, \dots, T \\
 & \sum_{j \in \{1, \dots, R\}} n_{tij} = n_{ti} \quad t = 1, \dots, T-1, i = 1, \dots, R \\
 & \sum_{j \in \{1, \dots, R\}} n_{tji} = n_{t+1,i} \quad t = 1, \dots, T-1, i = 1, \dots, R \\
 & n_{ti}, n_{tij} \in \mathbb{Z}_{\geq 0}
 \end{aligned}$$

「サンプルサイズ」と
「真の人口」の整合性の制約

例

時刻 1 時刻 2 時刻 3



時刻1の真の人口

エリア	人口
a	2
b	1

ここの和が
サンプルサイズ3に等しい

MAP 推定問題の数学的表現

$$\begin{aligned}
 \min_{\mathbf{n}} \quad & \sum_{t=1}^{T-1} \sum_{i,j \in \{1, \dots, R\}} (\log n_{tij}! - n_{tij} \log \phi_{tij}) + \sum_{t=2}^{T-1} \sum_{i \in \{1, \dots, R\}} (-\log n_{ti}!) \\
 & + \sum_{t=1}^T \sum_{i \in \{1, \dots, R\}} (-\log p_{ti}(y_{ti} | n_{ti})) \\
 \text{s. t.} \quad & \sum_{i \in \{1, \dots, R\}} n_{ti} = M \quad t = 1, \dots, T \\
 & \sum_{j \in \{1, \dots, R\}} n_{tij} = n_{ti} \quad t = 1, \dots, T-1, i = 1, \dots, R \\
 & \sum_{j \in \{1, \dots, R\}} n_{tji} = n_{t+1,i} \quad t = 1, \dots, T-1, i = 1, \dots, R \\
 & n_{ti}, n_{tij} \in \mathbb{Z}_{\geq 0}
 \end{aligned}$$

「真の人口」と
「真の移動人数」の
整合性の制約

例



時刻1の真の人口

エリア	人口
a	2
b	1

時刻1→2 の真の移動人数

	a	b
a	0	2
b	1	0

時刻2の真の人口

エリア	人口
a	1
b	2

MAP 推定問題の数学的表現

$$\begin{aligned} \min_{\mathbf{n}} \quad & \sum_{t=1}^{T-1} \sum_{i,j \in \{1, \dots, R\}} (\log n_{tij}! - n_{tij} \log \phi_{tij}) + \sum_{t=2}^{T-1} \sum_{i \in \{1, \dots, R\}} (-\log n_{ti}!) \\ & + \sum_{t=1}^T \sum_{i \in \{1, \dots, R\}} (-\log p_{ti}(y_{ti} | n_{ti})) \end{aligned}$$

$$\begin{aligned} \text{s. t.} \quad & \sum_{i \in \{1, \dots, R\}} n_{ti} = M & t = 1, \dots, T \\ & \sum_{j \in \{1, \dots, R\}} n_{tij} = n_{ti} & t = 1, \dots, T-1, i = 1, \dots, R \\ & \sum_{j \in \{1, \dots, R\}} n_{tji} = n_{t+1,i} & t = 1, \dots, T-1, i = 1, \dots, R \\ & n_{ti}, n_{tij} \in \mathbb{Z}_{\geq 0} \end{aligned}$$

} 人口・移動人数の整数性の制約

CGM の MAP 推定の既存手法

- 一般の CGM における MAP 推定問題は **NP-hard** [Sheldon et al., 2013]
 - 近似的な最適化問題を解くことで MAP 推定を行う手法が一般的
 - 以下を適用
 - ✓ **Stirling の近似** ($\log z! \approx z \log z - z$)
 - ✓ **連続緩和** (整数性制約を消去)
- ➡ 連続凸計画問題になるので
メッセージパッシング
などで解く [Sun et al., 2015]

CGM の MAP 推定の既存手法

- 一般の CGM における MAP 推定問題は **NP-hard** [Sheldon et al., 2013]
 - **近似的な最適化問題** を解くことで MAP 推定を行う手法が一般的
 - 以下を適用
 - ✓ **Stirling の近似** ($\log z! \approx z \log z - z$)
 - ✓ **連続緩和** (整数性制約を消去)
- ➡ 連続凸計画問題になるので
メッセージパッシング
などで解く [Sun et al., 2015]
- しかし, 近似を用いた最適化手法には問題点がいくつかある

① Stirling の近似は値が不正確な場合がある

- Stirling の近似: $\log z! \approx z \log z - z$ は **z が小さい場合に不正確**
- 最適化がうまくいかず**事後確率の小さい解**が出力されやすい

② 連続緩和の適用により整数解が得られない

- カウントの値が整数にならないのは困る

前半: 目次

- 研究の背景
- Collective Graphical Model (CGM)
- CGM における MAP 推定問題
- **提案手法**
- 実験

提案手法

連続緩和も Stirling の近似も用いない, 新しい MAP 推定の手法

- メリット

- 整数値制約も含む**全ての制約を満たす解**
- 既存手法よりも **事後確率の高い解**

- デメリット

- グラフィカルモデルのグラフが**パスグラフ**である場合にしか適用不可
(ただし, マルコフ連鎖に従う時系列を表現できるので重要)



パスグラフ

- アイデア

- 離散 **DC** (Difference of Convex) アルゴリズムの適用
- **最小凸費用流問題** への帰着

提案手法のアイデア ①

目的関数に着目

$$\min_n \sum_{t=1}^{T-1} \sum_{i,j \in \{1, \dots, R\}} (\log n_{tij}! - n_{tij} \log \phi_{tij}) + \sum_{t=2}^{T-1} \sum_{i \in \{1, \dots, R\}} (-\log n_{ti}!) \\ + \sum_{t=1}^T \sum_{i \in \{1, \dots, R\}} (-\log p_{ti}(y_{ti} | n_{ti}))$$

 の部分は一変数 **離散凸関数**, の部分は一変数 **離散凹関数**
であることを示せる

関数 $\mathcal{F}(z)$ が一変数 離散凸関数 $\Leftrightarrow \mathcal{F}(z+2) + \mathcal{F}(z) \geq 2 \cdot \mathcal{F}(z+1) \forall z \in \mathbb{Z}_{\geq 0}$

(**凸関数**) + (**凹関数**) の最小化問題には解くための方法論がある

➡ **DC アルゴリズム**

Difference of Convex (DC) アルゴリズム

離散 DC アルゴリズム [Maehara and Murota, 2015]

- (離散凸関数) + (離散凹関数) を最小化する方法
- 現在の解の付近で凹関数部分を線形近似した関数を計算 (代理関数)
→ 代理関数の最小化による解の更新
を繰り返す
- 目的関数は単調減少, 有限回の反復で停止
- 大域最適解が得られるとは限らないが,
経験的には良い解が得られることが知られている

Difference of Convex (DC) アルゴリズム

離散 DC アルゴリズム [Maehara and Murota, 2015]

- (離散凸関数) + (離散凹関数) を最小化する方法
- 現在の解の付近で凹関数部分を線形近似した関数を計算 (代理関数)
→ 代理関数の最小化による解の更新
を繰り返す

- 目的関数は単調減少, 有限回の反復で停止
- 大域最適解が得られるとは限らないが,
経験的には良い解が得られることが知られている

残された課題

代理関数が **効率的に最小化** できないと, 効率的なアルゴリズムにならない

代理関数の最小化問題

$$\begin{aligned}
 \min_{\mathbf{n}} \quad & \sum_{t=1}^{T-1} \sum_{i,j \in \{1, \dots, R\}} (\log n_{tij}! - n_{tij} \log \phi_{tij}) + \sum_{t=2}^{T-1} \sum_{i \in \{1, \dots, R\}} \left(-\log \left(n_{ti}^{(s)}! \right) + \alpha_{ti}^{(s)} \left(n_{ti} - n_{ti}^{(s)} \right) \right) \\
 & + \sum_{t=1}^T \sum_{i \in \{1, \dots, R\}} (-\log p_{ti}(y_{ti} | n_{ti})) \\
 \text{s. t.} \quad & \sum_{i \in \{1, \dots, R\}} n_{ti} = M \quad t = 1, \dots, T \\
 & \sum_{j \in \{1, \dots, R\}} n_{tij} = n_{ti} \quad t = 1, \dots, T-1 \\
 & \sum_{j \in \{1, \dots, R\}} n_{tij} = n_{t+1,i} \quad t = 1, \dots, T-1 \\
 & n_{ti}, n_{tij} \in \mathbb{Z}_{\geq 0}
 \end{aligned}$$

凹関数部分を線形近似

- $n_{ti}^{(s)}$: s ステップ目において求められた n_{ti} の値
- $\alpha_{ti}^{(s)}$: $-\log(n_{ti}^{(s)} + 1) \leq \alpha_{ti}^{(s)} \leq -\log n_{ti}^{(s)}$ をみたす実数

複雑な離散最適化問題 → 効率的に解くのが難しい

提案手法のアイデア②

代理関数の最小化問題は、**最小凸費用流問題** に帰着できる

最小凸費用流問題: 効率的に解ける組合せ最適化問題

入力

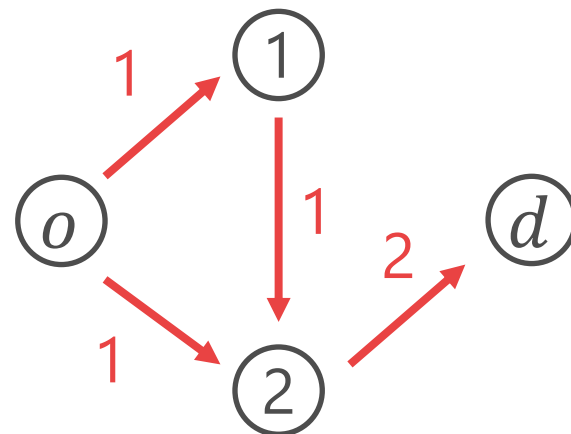
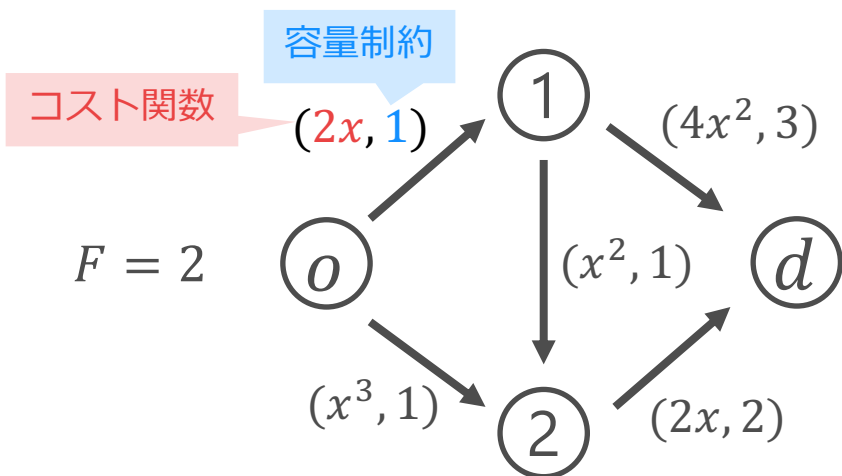
- ✓ 有向グラフ $G = (V, E)$
- ✓ 辺の容量制約 $u_{ij} \in \mathbb{Z}_{\geq 0}$,
- ✓ 辺のコスト関数 (一変数離散凸関数)

$$c_{ij}: \mathbb{Z}_{\geq 0} \rightarrow \mathbb{R}$$

- ✓ ソース頂点 o , シンク頂点 d , 総流量 $F \in \mathbb{Z}_{\geq 0}$

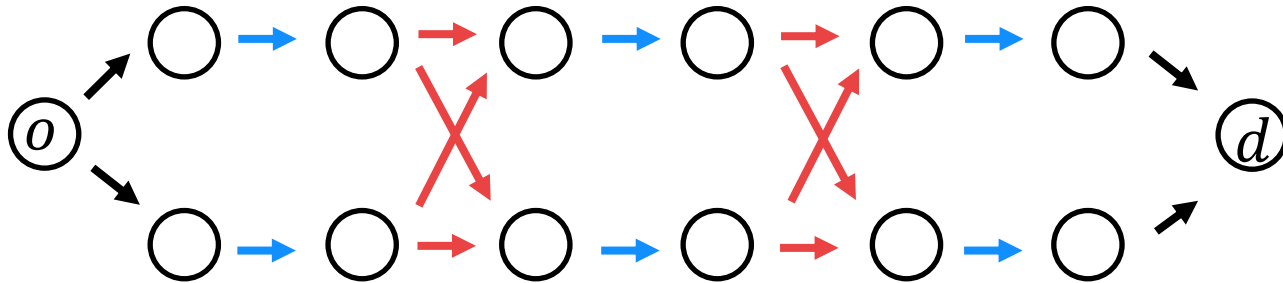
出力

- ✓ 各辺の容量制約を満たすような
 o から d への総流量 F のフローで,
総コストを最小化する もの



帰着: 代理関数最小化問題 → 最小凸費用流問題

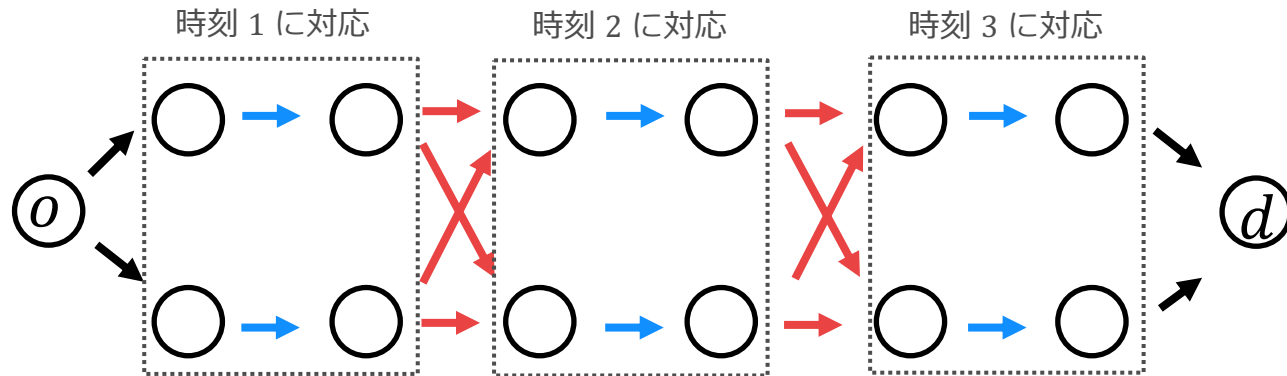
時刻数 3, エリア数 2 の場合



- トレリス図に似たネットワークを構築する
- 各辺に目的関数の各項をコスト関数として乗せる
- o から d に流量 M のフローを流す

帰着: 代理関数最小化問題 → 最小凸費用流問題

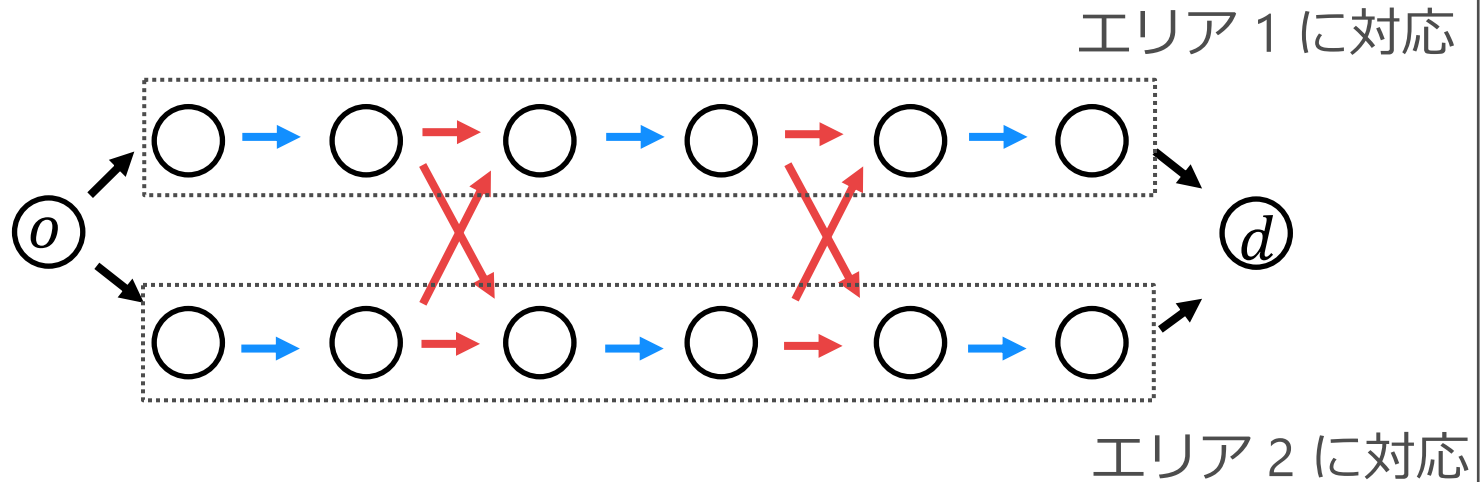
時刻数 3, エリア数 2 の場合



- トレリス図に似たネットワークを構築する
- 各辺に目的関数の各項をコスト関数として乗せる
- o から d に流量 M のフローを流す

帰着: 代理関数最小化問題 → 最小凸費用流問題

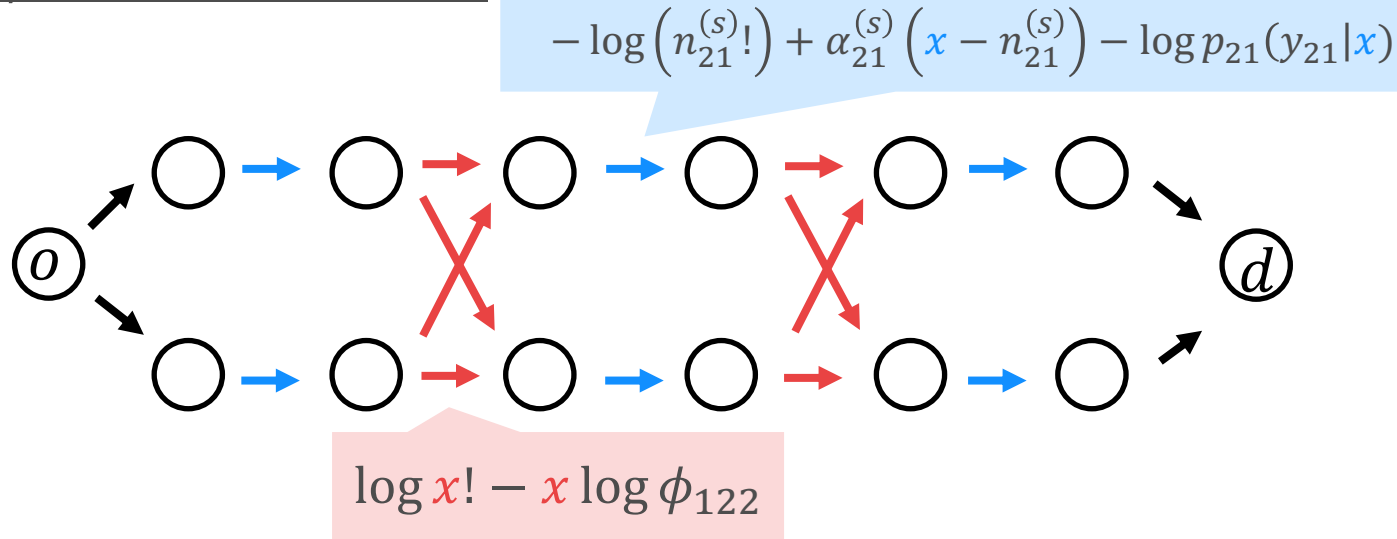
時刻数 3, エリア数 2 の場合



- トレリス図に似たネットワークを構築する
- 各辺に目的関数の各項をコスト関数として乗せる
- o から d に流量 M のフローを流す

帰着: 代理関数最小化問題 → 最小凸費用流問題

時刻数 3, エリア数 2 の場合



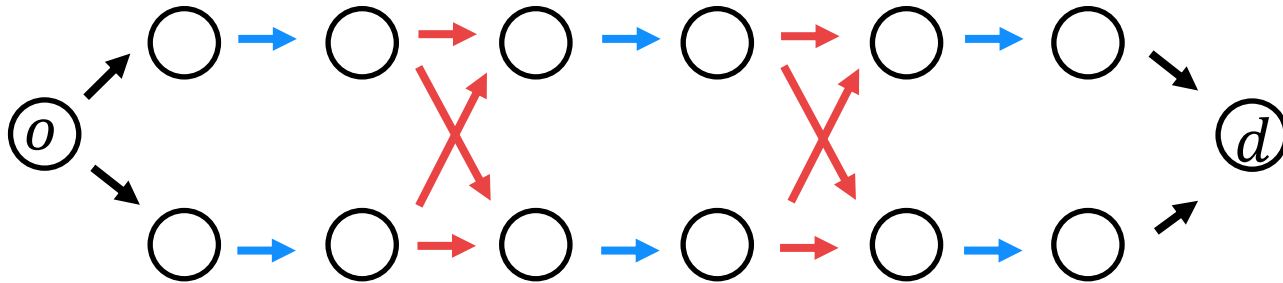
- 各辺に目的関数の各項をコスト関数として乗せる

目的関数

$$\sum_{t=1}^{T-1} \sum_{i,j \in \{1, \dots, R\}} (\log n_{tij}! - n_{tij} \log \phi_{tij}) + \sum_{t=2}^{T-1} \sum_{i \in \{1, \dots, R\}} \left(-\log(n_{ti}^{(s)}!) + \alpha_{ti}^{(s)}(n_{ti} - n_{ti}^{(s)}) \right) + \sum_{t=1}^T \sum_{i \in \{1, \dots, R\}} (-\log p_{ti}(y_{ti}|n_{ti}))$$

帰着: 代理関数最小化問題 → 最小凸費用流問題

時刻数 3, エリア数 2 の場合

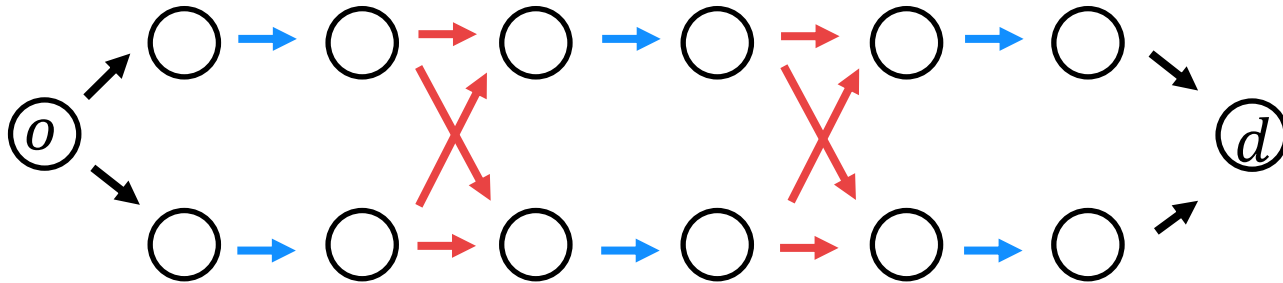


- トレリス図に似たネットワークを構築する
- 各辺に目的関数の各項をコスト関数として乗せる
- o から d に流量 M のフローを流す

- 「 \rightarrow を流れるフロー量」が「真の人口」を表現
- 「 \rightarrow を流れるフロー量」が「真の移動人数」を表現
- 「各頂点における流量保存則」が「制約」を表現

帰着: 代理関数最小化問題 → 最小凸費用流問題

時刻数 3, エリア数 2 の場合



Proposition

構築された最小凸費用流問題の最適解から,
代理関数の最小化問題の最適解を（高速に）計算可能



代理関数の最小化問題が効率的に解ける

最小凸費用流問題のアルゴリズム

最小凸費用流問題には色々な解き方がある
今回は 2 種類の手法を使用

- **Successive Shortest Path (SSP)**

ソース頂点からシンク頂点に最短路に沿ってフローを1ずつ流す手法
計算量 $O(MTR^2 \log(TR))$ → **エリア数・時刻数が大きい場合に速い**

- **Capacity Scaling (CS)**

大量のフローを一気に流す手法
計算量 $O(T^2 R^4 \log(TR) \log M)$ → **サンプルサイズが大きい場合に速い**

M : サンプルサイズ
 T : 時刻数
 R : エリア数

提案アルゴリズム

初期化 $\mathbf{n}^{(1)} \leftarrow \mathbf{0}$

for $s = 1, 2, \dots$, do

 現在の解 $\mathbf{n}^{(s)}$ をもとに最小凸費用流問題を構築

 最小凸費用流問題を解き, $\mathbf{n}^{(s+1)}$ を更新

 if $\mathcal{F}(\mathbf{n}^{(s+1)}) = \mathcal{F}(\mathbf{n}^{(s)})$ then $\mathcal{F}(\cdot)$: 目的関数

 break

 end if

end for

前半: 目次

- 研究の背景
- Collective Graphical Model (CGM)
- CGM における MAP 推定問題
- 提案手法
- **実験**

人工データを用いた実験

- MAP 推定問題をそれぞれの手法で解いた際に
達成される目的関数値（負の対数事後確率）・推定精度・計算速度 を比較
- 各サンプルサイズ M , 確率変数の値域のサイズ R に対して
ランダムなインスタンス10個を生成（時刻数: 5）
- 観測された人口 $\{y_{ti}\}_{t,i}$ から移動人数 $\{n_{tij}\}_{t,i,j}$ を推定
- 手法
 - ✓ Proposed: 提案手法
 - ✓ NLBP: Stirling の近似 + 連続緩和 + メッセージパッシング [Sun et al., 2015]
- グラフィカルモデルのポテンシャル ϕ_{tij} として2種類の設定を用意
 - ✓ Uniform: 各 ϕ_{tij} を $\{1, 5, 10\}$ の中からランダムで決定
 - ✓ Distance: $\phi_{tij} = \frac{1}{|i-j+1|}$ で決定
- 観測分布としてガウス分布を使用

目的関数値・推定誤差に関する結果

ポテンシャル: uniform

M		10^1			10^2			10^3		
R		10	20	30	10	20	30	10	20	30
Obj. Vals.	Proposed	-9.18e+01	-7.98e+01	-6.51e+01	-1.16e+03	-1.21e+03	-1.21e+03	-1.16e+04	-1.39e+04	-1.48e+04
	NLBP	-6.13e+01	-3.04e+01	-5.26e+00	-1.12e+03	-1.05e+03	-9.49e+02	-1.16e+04	-1.38e+04	-1.44e+04
NAE	Proposed	1.430	1.800	1.825	0.612	1.136	1.461	0.198	0.427	0.623
	NLBP	1.636	1.897	1.949	0.654	1.288	1.630	0.198	0.434	0.660

ポテンシャル: distance

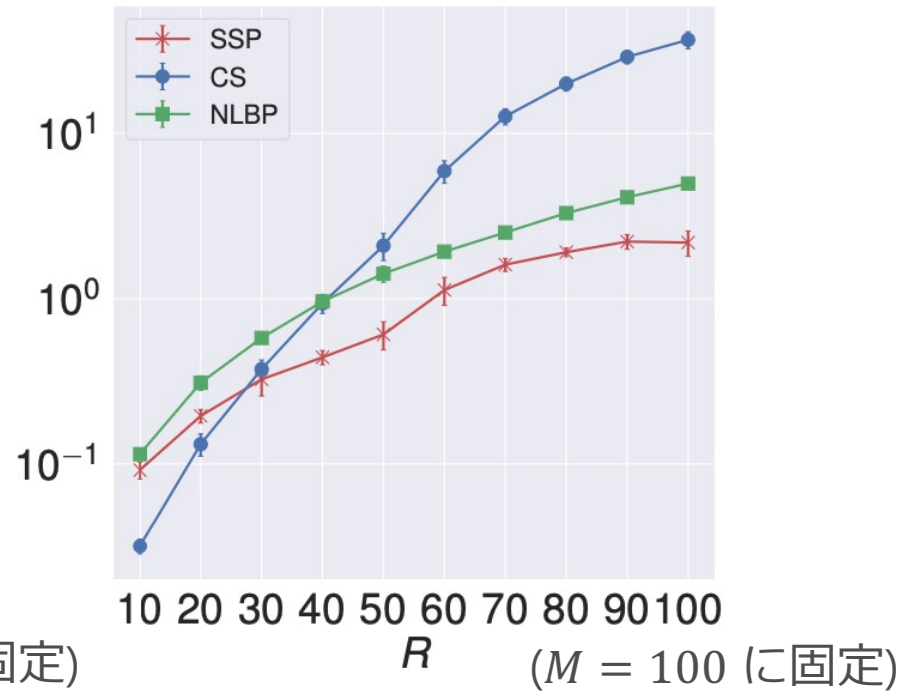
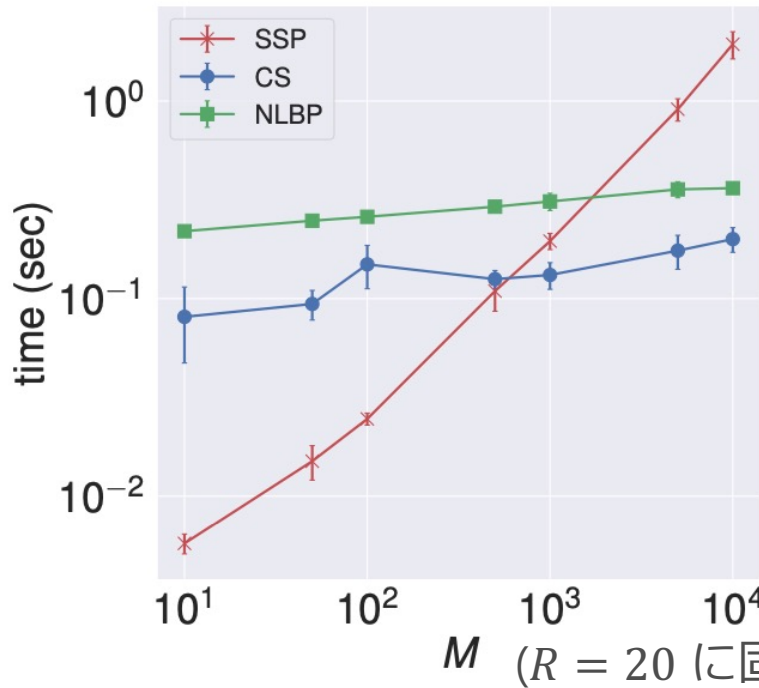
M		10^1			10^2			10^3		
R		10	20	30	10	20	30	10	20	30
Obj. Vals.	Proposed	1.72e+01	3.59e+01	5.29e+01	-4.99e+01	-6.46e+00	2.70e+01	-6.53e+02	-1.31e+03	-1.29e+03
	NLBP	5.28e+01	9.47e+01	1.29e+02	-4.25e+00	1.96e+02	3.58e+02	-6.36e+02	-1.19e+03	-7.99e+02
NAE	Proposed	1.495	1.715	1.825	0.666	1.141	1.324	0.209	0.436	0.648
	NLBP	1.633	1.845	1.916	0.678	1.254	1.501	0.208	0.431	0.653

Obj. Vals.: 目的関数値, NAE: 推定誤差 $\frac{\sum_{t=1}^T \sum_{i=1}^R \sum_{t=1}^R |n_{tij}^{\text{est}} - n_{tij}^{\text{true}}|}{\sum_{t=1}^T \sum_{i=1}^R \sum_{t=1}^R n_{tij}^{\text{true}}}$

- ほとんどのケースで提案手法は既存手法よりも **小さい目的関数値・推定誤差**
- 特に**サンプルサイズ M が小さい場合**,
エリア数 R が大きい場合 に差が大きくなる傾向

→変数の値が小さくなりやすく, Stirling の近似が不正確になりやすいため

計算速度に関する結果



- CS はサンプルサイズ M が大きくなっても高速
- SSP はエリア数 R が大きくなっても高速
- 問題のサイズ (M, R の大きさ) に応じて適切にアルゴリズムを選択することにより, 既存手法と同等の推定速度を達成可能

解の定性的な分析

- 二つの人口分布の間を補間（間の人口分布を欠損値として MAP 推定）
- 提案手法は **整数値での推定ができており, 0の要素が多い（スパース）**

既存手法は **解が非整数であり, ぼやけた解になっている**

入力

20	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
10	0	0	0	10

提案手法

$t = 1$

20	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

$t = 2$

3	3	0	0	0
8	6	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

$t = 3$

0	0	0	0	0
4	3	3	0	0
5	2	3	0	0
0	0	0	0	0
0	0	0	0	0

$t = 4$

0	0	0	0	0
0	0	0	0	0
4	1	3	3	0
5	0	3	1	0
0	0	0	0	0

$t = 5$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
6	1	0	5	2
3	0	0	2	1

$t = 6$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
10	0	0	0	10

既存手法

$t = 1$

19	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

$t = 2$

3.2	2.5	0.29	0	0
5.9	4.8	0.59	0	0
0.96	0.81	0.14	0	0
0	0	0	0.12	0
0	0	0	0	0

$t = 3$

0.59	0.63	0.31	0	0
2.7	2.8	1.7	0.33	0
3	3.1	2	0.44	0
0.62	0.66	0.47	0.23	0.12
0	0	0	0.12	0

$t = 4$

0.16	0.21	0.11	0	0
0.75	0.64	0.52	0.34	0
2.8	2	2	1.5	0.26
2.3	1.6	1.6	1.4	0.34
0.34	0.25	0.28	0.34	0.17

$t = 5$

0.11	0.15	0	0	0
0.16	0.2	0.13	0	0
0.78	0.39	0.18	0.55	0.26
4.1	1.6	0.59	3.2	1.7
2	0.76	0.28	1.7	1

$t = 6$

0.12	0.12	0	0	0
0.12	0.13	0.1	0	0
0	0.1	0	0	0
0	0	0	0	0
9.4	0	0	0	9.2

まとめ

- 集計された人の位置情報データを扱うための確率モデルである **Collective Graphical Model** における **MAP 推定問題**
- **Stirling の近似及び連続緩和を用いない新しい手法** を提案
 - ✓ アイデア: 離散 DC アルゴリズム + 最小凸費用流
- 人工データ・実データを用いて提案手法の有効性を検証



現在バイアスの影響下における 解析的に扱いやすい人間の行動モデル

Yasunori Akagi, Naoki Marumo, Takeshi Kurashima.

“Analytically Tractable Models for Decision Making under Present Bias.”

Preprint (arxiv: <https://arxiv.org/abs/2302.08132>)

- **研究の背景**
- 提案モデル
- エージェントの状態系列の解析解
- 提案モデルの解析
 - ✓ タスク放棄と現在バイアスの関係
 - ✓ 目標の最適化問題
 - ✓ 報酬スケジュールの最適化問題

人間は、現在の利得や損失を過大評価してしまうバイアス
（**現在バイアス**）を持っており、
長期的な目標達成 や **効率的な計画遂行** を阻害することがある

例1. ダイエット

- ダイエットをしたいと決意して、8月までに10kg痩せる目標を立てる
- 金曜日の夜になると、直近の「飲み会に行く利得」が大きく評価され、
「8月に痩せている利得」を上回る
- 結果、飲み会で暴飲暴食、ダイエット失敗…



人間は、現在の利得や損失を過大評価してしまうバイアス
（**現在バイアス**）を持っており、
長期的な目標達成 や **効率的な計画遂行** を阻害することがある

例2. 夏休みの宿題

- 夏休みの最初は、毎日コツコツ計画的に宿題を進めることを計画
- しかし、いざ宿題をする時間になると
「宿題をするコスト」が大きく評価され、
先延ばししてしまう
- 結局、最終日に全て終わらせるハメに…



やりたいこと

- 現在バイアスを **数理的にモデリング** し,
現在バイアスが人の行動に与える影響 を調べる
- 数理モデルに基づく **適切な介入** を行うことで
現在バイアスによる計画の頓挫を防ぎ, 目標達成に導く

介入の例

- 適切な目標を設定する
 - 中間的な目標・報酬を与える
 - 選択肢を追加・制限する
- など…

- 現在バイアス及びその数理モデルについては、行動経済学分野において古くから研究されている

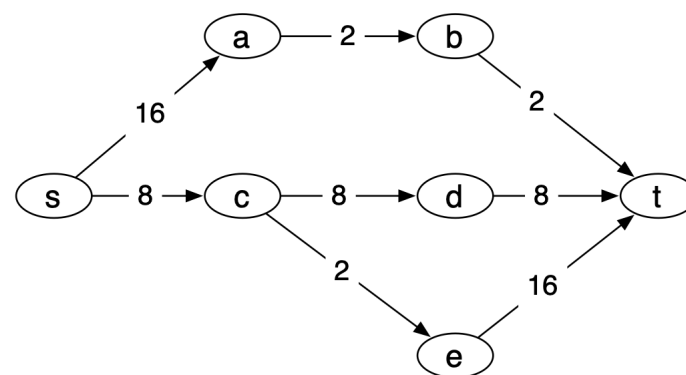
代表例

- Samuelson (1937): 割引効用モデル (指数割引)
 - Ainslie (1975): 双曲割引モデル
 - Laibson (1997): 準双曲割引モデル
 - O'Donoghue and Rabin (1999): 先延ばしと現在バイアスのモデル
- 近年, **グラフを用いたモデル** が提案され, 注目されている
[Kleinberg and Oren, EC'14] [Kleinberg et al., EC'16]
[Kleinberg et al., EC'17]

グラフを用いたモデル (KO モデル)

現在バイアス下の人がタスクを行う様子を, **有向非巡回グラフ** でモデル化

- 有向非巡回グラフ $G = (V, E)$
 - ✓ 頂点 \leftrightarrow 人間の状態
 - ✓ 辺 \leftrightarrow 人間の行動
- 辺のコスト $c_{ij} \in \mathbb{R}_{\geq 0}$
- スタート頂点 s , ゴール頂点 t



[Kleinberg and Oren, EC'14] より

グラフを用いたモデル (KO モデル)

現在バイアス下の人がタスクを行う様子を, **有向非巡回グラフ** でモデル化

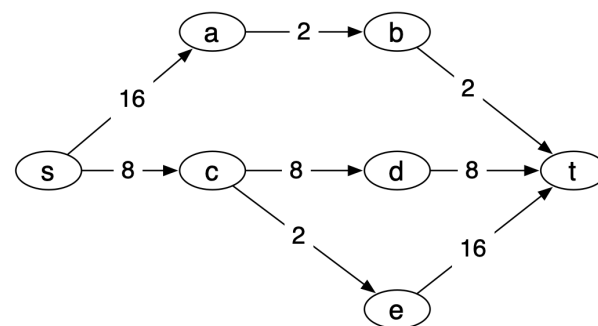
このグラフ上で, 人は以下のルールで s から t に移動する

- 人は今いる頂点において,
「 t までのパスの中で **現在バイアスの影響を考慮したコスト** が最小になるもの」
を選び, そのパスを辿って一つ枝を移動する

- 現在バイアスの影響を考慮したコスト** は,
パス $u_1, u_2, \dots, u_m = t$ に対して

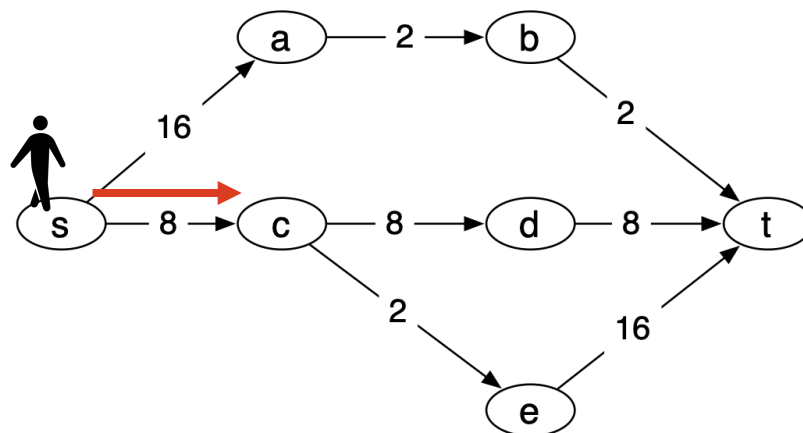
$$\frac{1}{\beta} \cdot c_{u_1, u_2} + \sum_{i=2}^{m-1} c_{u_i, u_{i+1}}$$

※ $\beta \in (0, 1]$: 現在バイアスパラメータ



KO モデルにおける動きの例

$\beta = \frac{1}{2}$ の場合



$s \rightarrow a \rightarrow b \rightarrow t$: コストは $2 \cdot 16 + 2 + 2 = 36$

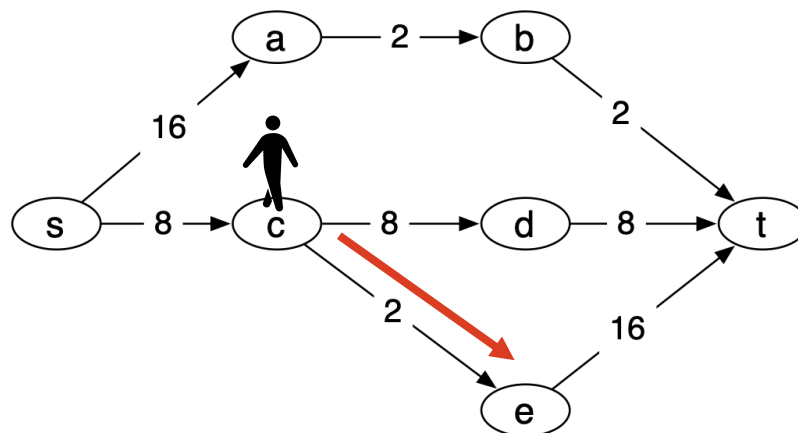
$s \rightarrow c \rightarrow d \rightarrow t$: コストは $2 \cdot 8 + 8 + 8 = 32$

$s \rightarrow c \rightarrow e \rightarrow t$: コストは $2 \cdot 8 + 2 + 16 = 34$

$s \rightarrow c \rightarrow d \rightarrow t$ のコストが
最小なので, $s \rightarrow c$ と移動

KO モデルにおける動きの例

$\beta = \frac{1}{2}$ の場合



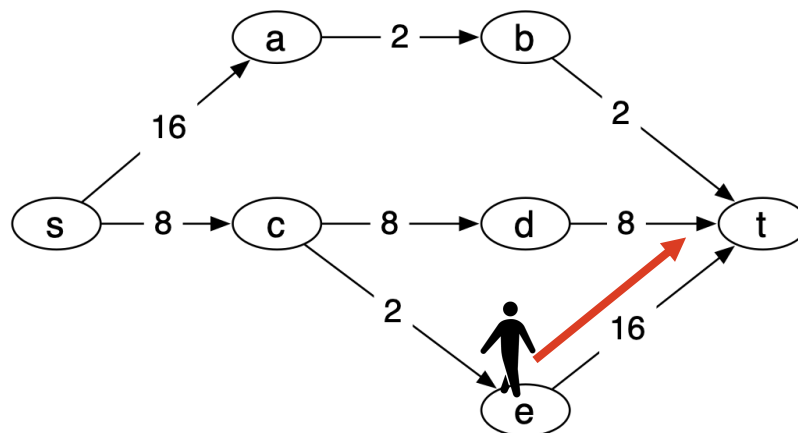
$c \rightarrow d \rightarrow t$: コストは $2 \cdot 8 + 8 = 24$

$c \rightarrow e \rightarrow t$: コストは $2 \cdot 2 + 16 = 20$

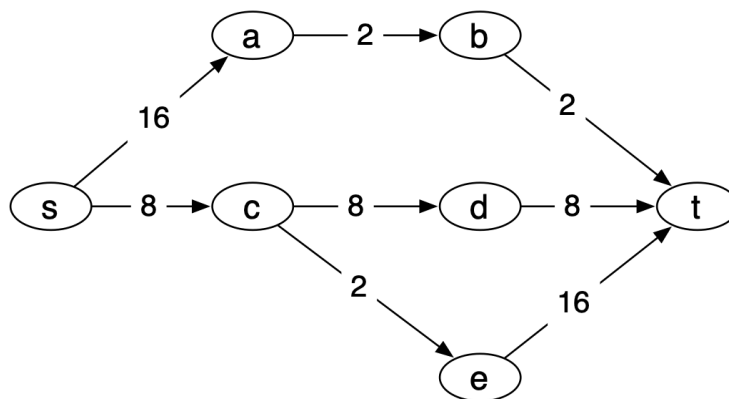
} $c \rightarrow e \rightarrow t$ のコストが
最小なので, $c \rightarrow e$ と移動

KO モデルにおける動きの例

$\beta = \frac{1}{2}$ の場合



パスは $e \rightarrow t$ しかないので $e \rightarrow t$ と移動



- $s \rightarrow t$ 最短路は $s \rightarrow a \rightarrow b \rightarrow t$ でコスト 20
実際に辿ったパスは $s \rightarrow c \rightarrow e \rightarrow t$ でコスト 26
→ 現在バイアスの影響により, **コストの大きいパス** を選ばされている
- s にいるときは「 $c \rightarrow d \rightarrow t$ 」というパスを想定しているのに,
実際に c にいるときは「 $c \rightarrow e \rightarrow t$ 」というパスで動く
→ 以前の計画時とは異なる行動を取ってしまう (**Time Inconsistency**)

- 先延ばし, タスク放棄などの **現在バイアスに由来する現象をよく再現**
- 様々な現実世界での目標達成行動をモデル化する **高い表現力**

により注目され, **様々な解析・後続研究**

例: 初出の論文 [Kleinberg and Oren, 2014] での結果

- コスト比 (最短路のコストと実際に人がとるパスのコストの比)
が指数的に小さくなるグラフの, グラフマイナーによる特徴づけ
- 現在バイアスパラメータを変化させたときの, パスの種類数のバウンド
- 人がゴールできる最小部分グラフの構造の特徴づけ など

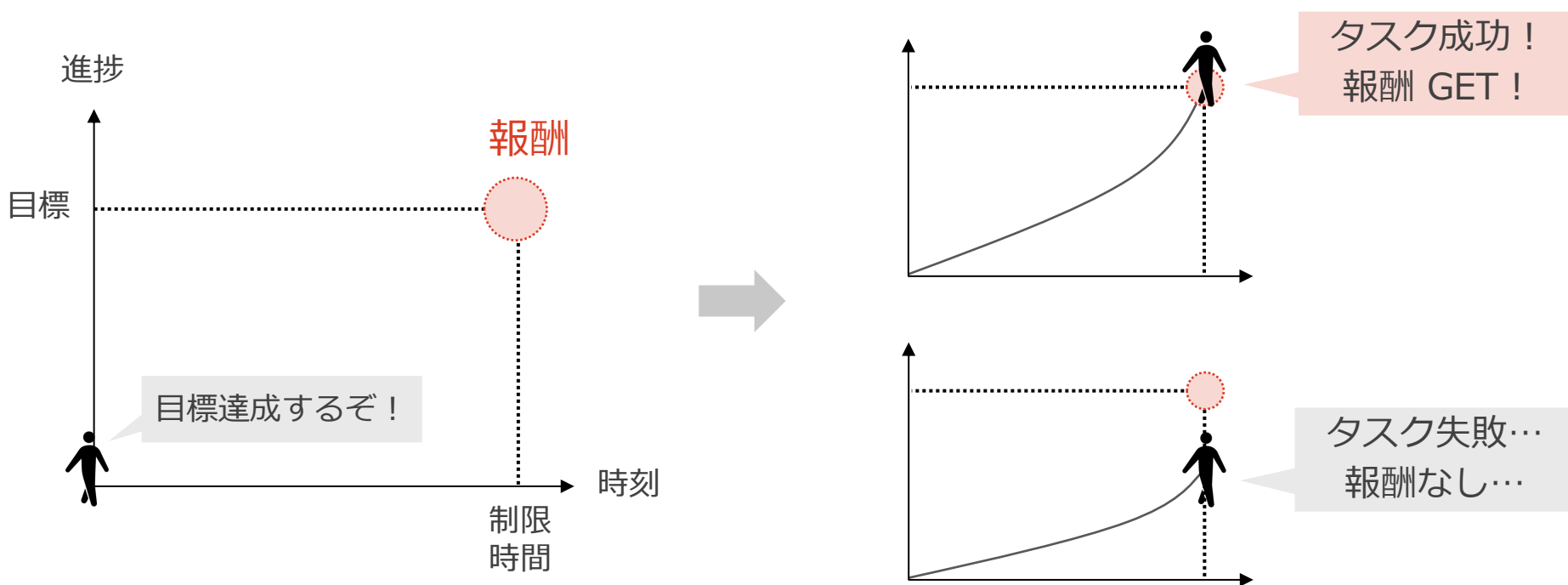
- 研究の背景
- **提案モデル**
- エージェントの状態系列の解析解
- 提案モデルの解析
 - ✓ タスク放棄と現在バイアスの関係
 - ✓ 目標の最適化問題
 - ✓ 報酬スケジュールの最適化問題

- モデルの性質の解析が難しい
 - ✓ エージェントの行動を閉形式で書き表すことができず,
モデルのパラメータを変更した時の挙動の変化を理論的に解析するのが難しい
 - ✓ 実際の行動を知るためにはシミュレーションを行う必要がある
- エージェントをゴールに導くための最適な介入を求めるのが難しい
 - ✓ 中間報酬を適切に配置する介入 [Tang et al., 2017] [Albers and Kraft, 2019]
 - ✓ 辺のコストを減らす介入 [Albers and Kraft, 2021]
などが提案されているが,
いずれも **最適な介入を求めるのは NP 困難**

- 現在バイアスの影響下の人間の行動をモデリングするための新しいモデルを提案
 - ✓ KO モデルがベース
 - ✓ 「進捗積み上げ型タスク」を扱う
 - ✓ エージェントのとり状態の系列を
閉形式で書き表すことができる という著しい特長を持つ
- 提案モデルを用いて、3つの重要な問題を解析
 - ✓ タスク放棄と現在バイアスの関係
 - ✓ 目標の最適化問題
 - ✓ 報酬スケジュールの最適化問題

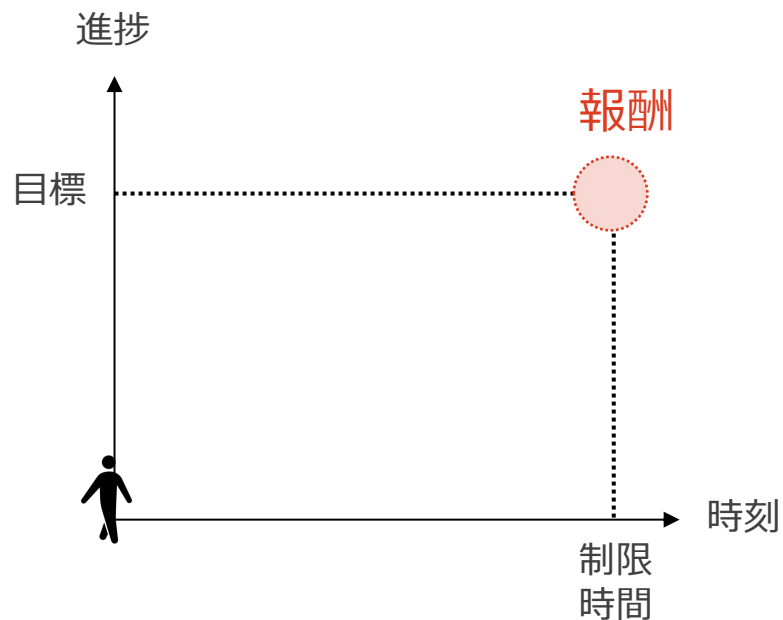
進捗積み上げ型タスク

制限時間の間に「進捗」を積み上げて「目標」の達成を目指し、
達成すると「報酬」が得られるようなタスク



進捗積み上げ型タスク

制限時間の間に「進捗」を積み上げて「目標」の達成を目指し、
達成すると「報酬」が得られるようなタスク



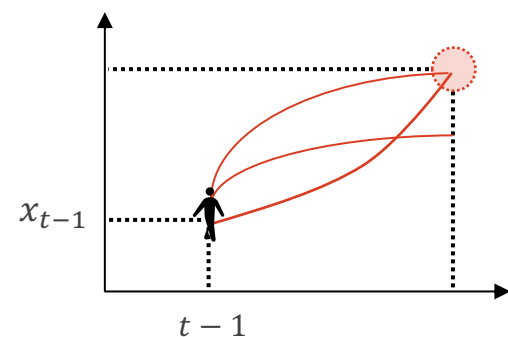
このようなタスクは
日常生活の中でも頻繁に現れる

- 健康のために1週間で10時間運動する
- 6ヶ月で卒業論文を完成させる
- 1年間で100万ドルの売上目標を達成する

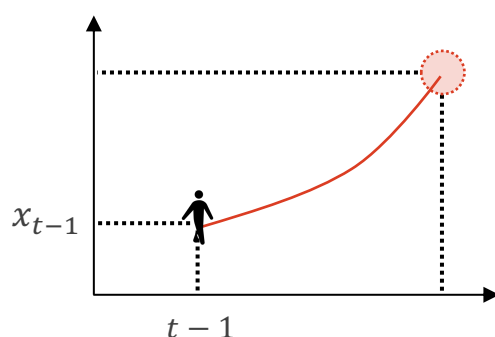
状態 $(t-1, x_{t-1})$ にいる人は、以降の状態の系列 $(t, y_t), (t+1, y_{t+1}), \dots, (T, y_T)$ の **総コスト** を

$$\frac{1}{\beta} c(y_t - x_{t-1}) + \sum_{i=t+1}^T c(y_i - y_{i-1}) - R \cdot \mathbf{1}[y_T \geq \theta]$$

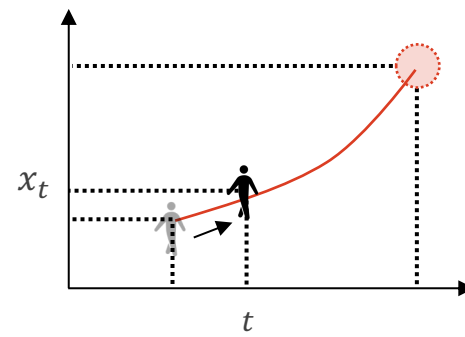
で評価し、これを最小にするような $y_t^*, y_{t+1}^*, \dots, y_T^*$ を求めて $(t-1, x_{t-1}) \rightarrow (t, y_t^*)$ と移動



今後の移動系列の候補
(赤線) の中で、
総コストを比較



総コスト最小のパスが
見つかったら、



それに沿って移動

状態 $(t-1, x_{t-1})$ にいる人は、以降の状態の系列 $(t, y_t), (t+1, y_{t+1}), \dots, (T, y_T)$ の **総コスト** を

$$\underbrace{\frac{1}{\beta} c(y_t - x_{t-1})}_{\text{直近かかるコスト}} + \underbrace{\sum_{i=t+1}^T c(y_i - y_{i-1}) - R \cdot \mathbf{1}[y_T \geq \theta]}_{\text{将来かかるコスト}}$$

$c(\Delta)$: 進捗 Δ を生むために必要なコスト

$$c(\Delta) = \begin{cases} \Delta^\alpha & \Delta \geq 0 \\ +\infty & \Delta < 0 \end{cases} \quad \text{の形を仮定 } (\alpha > 1)$$

- コスト関数の凸性, 単調増加性など望ましい性質を備える形
- $\Delta < 0 \Rightarrow +\infty$ は, 進捗が単調増加であることを表現

状態 $(t-1, x_{t-1})$ にいる人は、以降の状態の系列 $(t, y_t), (t+1, y_{t+1}), \dots, (T, y_T)$ の **総コスト** を

$$\underbrace{\frac{1}{\beta} c(y_t - x_{t-1})}_{\text{直近かかるコスト}} + \underbrace{\sum_{i=t+1}^T c(y_i - y_{i-1}) - R \cdot \mathbf{1}[y_T \geq \theta]}_{\text{将来かかるコスト}}$$

β : 現在バイアスの強さを表すパラメータ ($0 < \beta \leq 1$)

➤ 直近かかるコストのみ $1/\beta$ がかかって増幅されている

→ 現在バイアスの効果を表現

➤ β が大きい → 増幅が小さいので現在バイアスは弱い

➤ β が小さい → 増幅が大きいので現在バイアスは強い

状態 $(t-1, x_{t-1})$ にいる人は、以降の状態の系列 $(t, y_t), (t+1, y_{t+1}), \dots, (T, y_T)$ の **総コスト** を

$$\frac{1}{\beta} c(y_t - x_{t-1}) + \sum_{i=t+1}^T c(y_i - y_{i-1}) - \underline{R \cdot \mathbf{1}[y_T \geq \theta]}$$

報酬に関する項

- R : 報酬額
- θ : 目標となる進捗
- 報酬に関する項は,
「最終進捗 y_T が目標進捗 θ を上回れば報酬 R が得られる」
ことを意味する

- 研究の背景
- 提案モデル
- **エージェントの状態系列の解析解**
- 提案モデルの解析
 - ✓ タスク放棄と現在バイアスの関係
 - ✓ 目標の最適化問題
 - ✓ 報酬スケジュールの最適化問題

- 時刻 t の進捗 x_t は以下の式に従って決定

$$x_t = \operatorname{argmin}_{y_t \in \mathbb{R}} \min_{y_{t+1}, \dots, y_T \in \mathbb{R}} \left\{ \frac{1}{\beta} c(y_t - x_{t-1}) + \sum_{i=t+1}^T c(y_i - y_{i-1}) - R \cdot \mathbf{1}[y_T \geq \theta] \right\}$$

- 複雑なダイナミクスになりそうだが, Jensen・Hölder の不等式をうまく利用することで以下の **漸化式** が成り立つことを示せる

$$x_t = \begin{cases} \theta + p_t(x_{t-1} - \theta) & \text{if } x_{t-1} \geq \theta - R^{\frac{1}{\alpha}} \left(T - t + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \\ x_{t-1} & \text{otherwise} \end{cases}$$

$$\text{ただし } p_t = \frac{T-t}{T-t+\beta^{\frac{1}{\alpha-1}}}$$

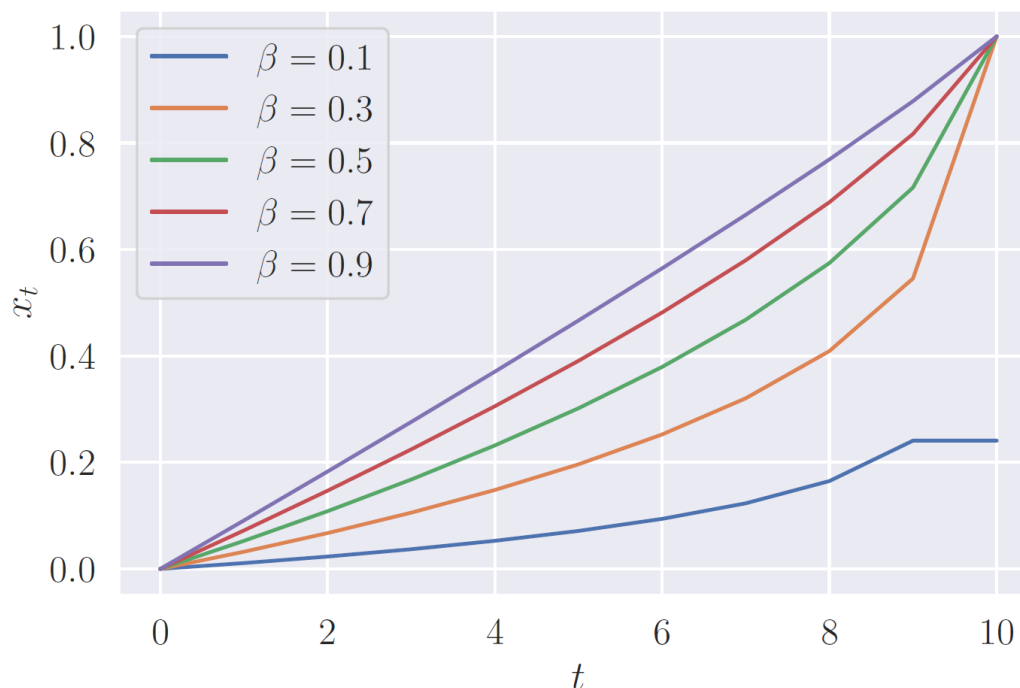
漸化式より，人のとる挙動を数学的に閉形式で記述できる

$$\text{時刻 } t \text{ における進捗 } x_t = \begin{cases} \theta \left(1 - \prod_{i=1}^t p_i \right) & \text{if } t \leq t^* \\ \theta \left(1 - \prod_{i=1}^{t^*} p_i \right) & \text{otherwise} \end{cases}$$

ただし t^* は以下を満たす最小の $t \in \{0, \dots, T-1\}$

$$\left(T - t - 1 + \beta^{\frac{1}{\alpha-1}} \right)^{1-\alpha} \prod_{i=1}^t p_i^\alpha > \frac{R}{\theta^\alpha}$$

→ モデルの性質の解析や最適な介入策の導出 を容易に行える！



- β が大きいとき（現在バイアスが **弱い** とき）→ **コツコツ進捗を生む**
- β が小さいとき（現在バイアスが **強い** とき）→ **先延ばしにして最後の一気にやる**
（途中で諦めてしまう場合も…）

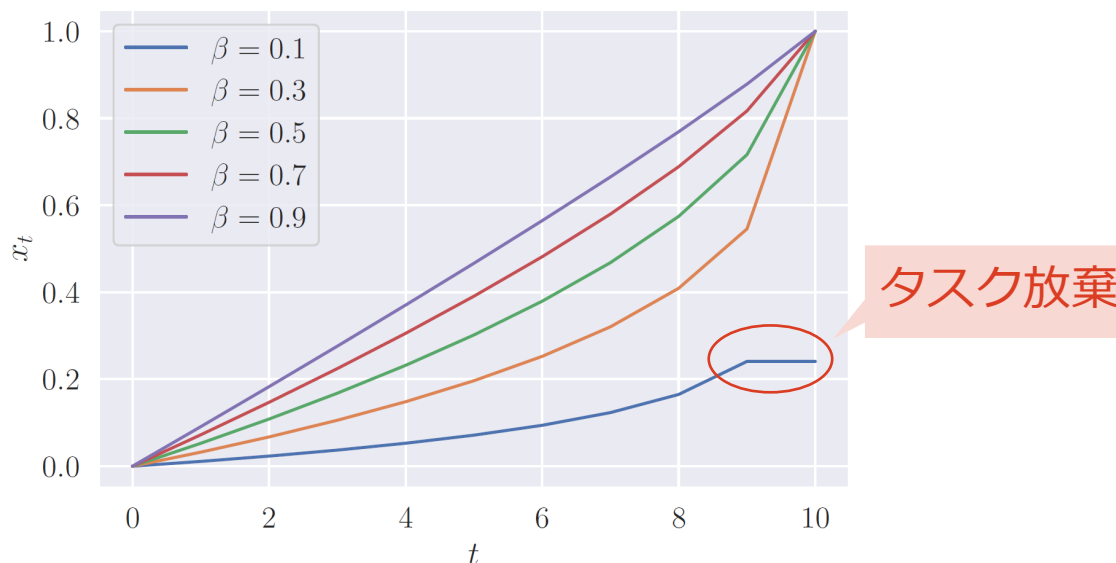
- 研究の背景
- 提案モデル
- エージェントの状態系列の解析解
- 提案モデルの解析
 - ✓ **タスク放棄と現在バイアスの関係**
 - ✓ 目標の最適化問題
 - ✓ 報酬スケジュールの最適化問題

現在バイアスが強いと途中でタスクを放棄してしまうことがある

直近のコストを嫌がって進捗を積むことを先延ばし

→ あるところでタスク達成の報酬が必要な努力に見合わなくなる

→ タスク放棄



現在バイアスとタスク放棄の関係は？

定義

現在バイアスパラメータ β を持つエージェントについて,
ある θ, R が存在して
タスクを諦める時刻 t^* が $t^* \neq 0, T$ を満たすとき,
 β は **Task-Abandonment Inducing (TAI)** であるという

- $t^* = 0 \rightarrow$ タスクを初めからやらない
- $t^* = T \rightarrow$ タスクを完遂し目標を達成する
- $t^* \neq 0, T \rightarrow$ **タスクを途中までやって放棄する**

TAI \Leftrightarrow 目標や報酬によってはタスクの **途中放棄を行う可能性がある**

TAI でない \Leftrightarrow タスクの途中放棄を **決して行わない**

定理

T と α によってのみ決まり, $\left(1 - \frac{1}{\alpha}\right)^{\alpha-1} < \beta_0 < \left(1 - \frac{1}{\alpha}\right)^{\frac{\alpha-1}{2}}$ を満たすある定数 β_0 が存在し,

$$\beta \text{ が TAI} \Leftrightarrow \beta < \beta_0$$

※ α はコスト関数の形状を決めるパラメータ ($c(\Delta) = \Delta^\alpha$)

- TAI である必要十分条件を具体的に書き表すことができた
- β がある程度 **小さい** (現在バイアスが強い) と TAI になる
- $\left(1 - \frac{1}{\alpha}\right)^{\alpha-1} > \frac{1}{e} \approx 0.368 \dots$ (ただし e は自然対数の底) が成り立つので,
 $\beta < \frac{1}{e}$ のときは α の値に関係なく TAI である

定理

$$\beta_0 = \left(1 - \frac{1}{\alpha}\right)^{\alpha-1} \left(1 + \frac{\alpha \log \Gamma\left(1 - \frac{1}{\alpha}\right) + \log\left(1 - \frac{1}{\alpha}\right)}{\log T}\right) + o\left(\frac{1}{\log T}\right)$$

ただし $\Gamma(\cdot)$ はガンマ関数

- $T \rightarrow \infty$ のとき, $\beta_0 \rightarrow \left(1 - \frac{1}{\alpha}\right)^{\alpha-1}$
- ただし収束の速さは, $\beta_0 - \left(1 - \frac{1}{\alpha}\right)^{\alpha-1} = \Theta\left(\frac{1}{\log T}\right)$ なのでかなり遅い
- $\alpha = 2$ (二乗コスト) の場合,

$$\beta = \frac{1}{2} + \frac{\log\left(\frac{\pi}{2}\right)}{2 \log T} + o\left(\frac{1}{\log T}\right) \text{ となり, なぜか (?) 円周率が出てくる}$$

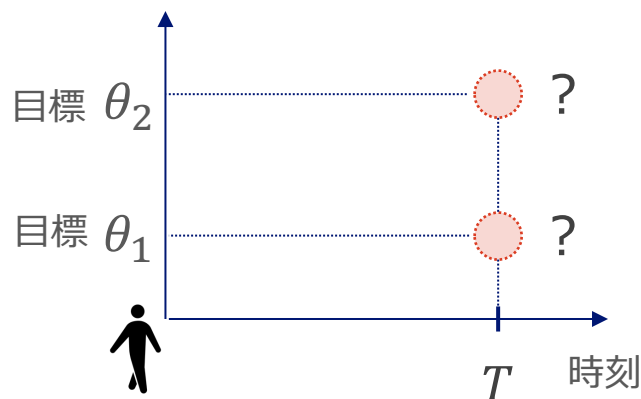
- 研究の背景
- 提案モデル
- エージェントの状態系列の解析解
- 提案モデルの解析
 - ✓ タスク放棄と現在バイアスの関係
 - ✓ **目標の最適化問題**
 - ✓ 報酬スケジュールの最適化問題

介入①: 目標の最適化問題

総期間 T と報酬 R が与えられたとき,

どのような目標 θ を設定すれば達成される進捗を最大化できるか？

$$\max_{\theta \geq 0} x_T$$



目標をうまく設定して
この人の進捗を
最大化するぞ！

- 目標が低すぎると, 小さな進捗で満足してしまうため, 達成される進捗が小さく
- 目標が高すぎると, 途中で諦めてしまうため, 達成される進捗が小さく

目標の最適化問題: 最適解

結果: 最適な目標設定を導出

(1) $\beta \geq \beta_0$ のとき $\theta = R^{\frac{1}{\alpha}} \left(T - 1 + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}}$ が最適.

(2) $\beta < \beta_0$ のとき $\theta = R^{\frac{1}{\alpha}} \left(T - t^* + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \frac{\Gamma(T - t^* + 1) \Gamma\left(T + \beta^{\frac{1}{\alpha-1}}\right)}{\Gamma(T) \Gamma\left(T - t^* + 1 + \beta^{\frac{1}{\alpha-1}}\right)}$ が最適.

ただし $t^* := \operatorname{argmax}_{t \in \{1, \dots, T\}} u_t$

$$u_t := \left(T - t + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \left(\frac{\Gamma(T - t + 1) \Gamma\left(T + \beta^{\frac{1}{\alpha-1}}\right)}{\Gamma(T) \Gamma\left(T - t + 1 + \beta^{\frac{1}{\alpha-1}}\right)} - \frac{T - t}{T - t + \beta^{\frac{1}{\alpha-1}}} \right)$$

$\Gamma(\cdot)$ はガンマ関数

目標の最適化問題: 最適解

結果: 最適な目標設定を導出

時間計算量 $O(1)$ で計算可能!

(1) $\beta \geq \beta_0$ のとき $\theta = R^{\frac{1}{\alpha}} \left(T - 1 + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}}$ が最適.

(2) $\beta < \beta_0$ のとき

$\theta = R^{\frac{1}{\alpha}} \left(T - t^* + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \frac{\Gamma(T - t^* + 1) \Gamma\left(T + \beta^{\frac{1}{\alpha-1}}\right)}{\Gamma(T) \Gamma\left(T - t^* + 1 + \beta^{\frac{1}{\alpha-1}}\right)}$ が最適.

時間計算量
 $O(T)$ で計算可能!

ただし $t^* := \operatorname{argmax}_{t \in \{1, \dots, T\}} u_t$

$$u_t := \left(T - t + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}} \left(\frac{\Gamma(T - t + 1) \Gamma\left(T + \beta^{\frac{1}{\alpha-1}}\right)}{\Gamma(T) \Gamma\left(T - t + 1 + \beta^{\frac{1}{\alpha-1}}\right)} - \frac{T - t}{T - t + \beta^{\frac{1}{\alpha-1}}} \right)$$

$\Gamma(\cdot)$ はガンマ関数

目標の最適化問題: 搾取的な報酬

- $\beta < \beta_0$ のとき,
「時刻 $t < T$ でタスク放棄するように目標を設定する」ことが最適になる場合がある
この場合, 頑張って進捗を積んだのに報酬を受け取ることができない
(**搾取的な報酬**)
- $\beta \geq \beta_0$ のときは搾取的な報酬は有効ではなく,
常にタスク放棄させないようにする のが最適になる



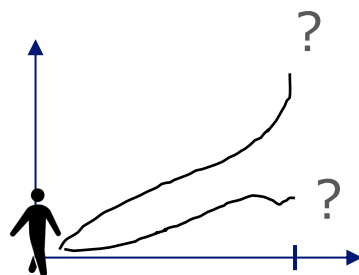
現在バイアスが強い人は,
悪意のある介入者によって搾取されることがある
現在バイアスが弱い人は,
搾取されにくい

- 研究の背景
- 提案モデル
- エージェントの状態系列の解析解
- 提案モデルの解析
 - ✓ タスク放棄と現在バイアスの関係
 - ✓ 目標の最適化問題
 - ✓ **報酬スケジュールの最適化問題**

介入②: 最適な報酬スケジューリング

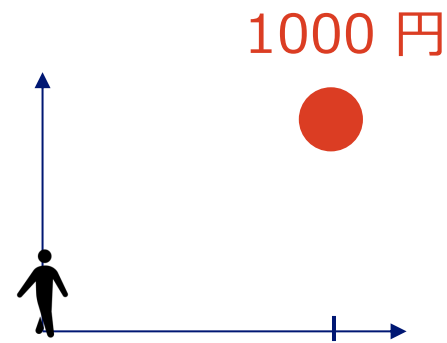
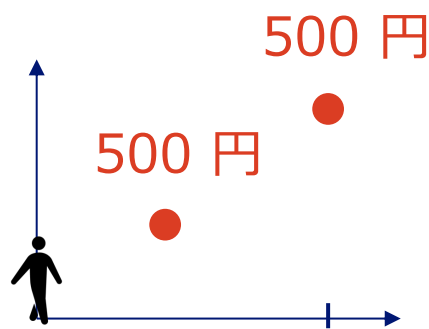
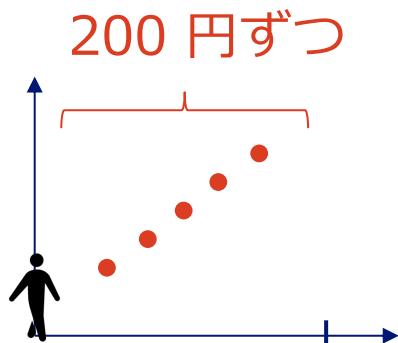
総期間 T と総報酬 R が与えられたとき,

どのような報酬スケジュールが人の進捗を最大化するか？



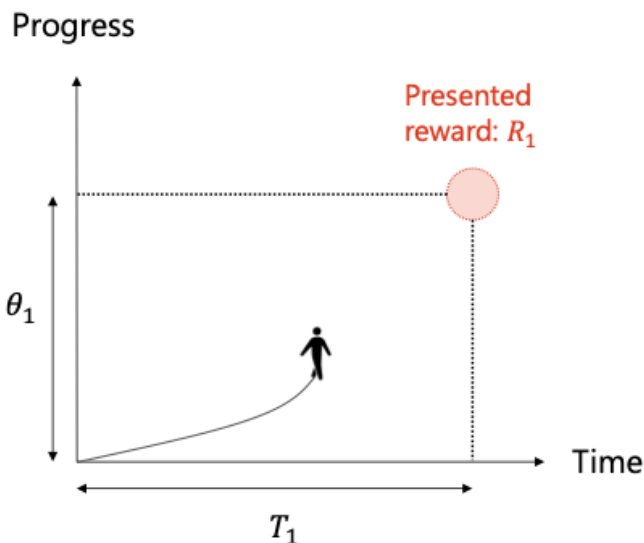
報酬をうまく与えて
被験者の進捗を
最大化するぞ！

報酬の与え方は色々考えられる

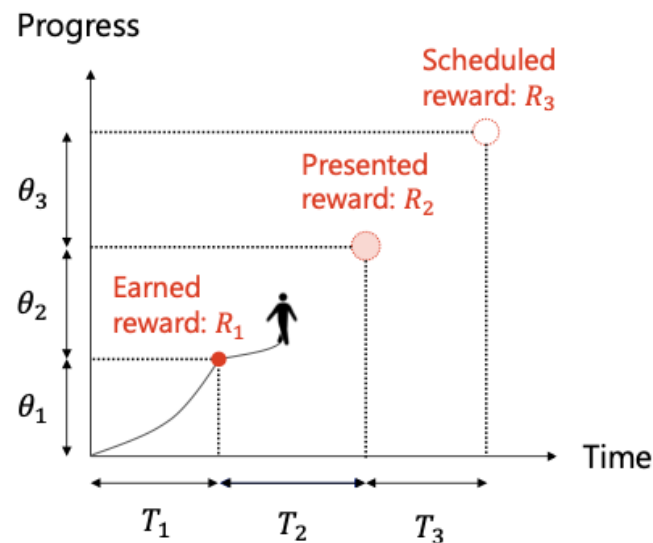


最適な報酬スケジューリング: 定式化

- 分割回数 $k \in \{1, \dots, T\}$ を決め,
総期間を $T_1, \dots, T_k \in \mathbb{Z}_{>0}$, 総報酬を R_1, \dots, R_k に分割する
- エージェントは, i 番目の期間 T_i において
「進捗を $\theta_i \in \mathbb{R}_{\geq 0}$ 増やせば報酬 R_i を得る」という条件で進捗を積む
- 適切に $k, (T_i)_{i=1}^k, (R_i)_{i=1}^k, (\theta_i)_{i=1}^k$ を定めて最終的な進捗を最大化する



(a) $k = 1$



(b) $k = 3$

※ 搾取的な報酬は禁止

最適な報酬スケジューリング: 結果

結果: 最適な報酬スケジュールを導出

(1) $\beta \geq \beta_0$ のとき, 最適な報酬スケジュールは

$$k = 1, T_1 = T, R_1 = R, \theta_1 = R^{\frac{1}{\alpha}} \left(T - 1 + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}}$$

(2) $\beta < \beta_0$ のとき, 最適な報酬スケジュールは

$$k = k^*, T_i = T_i^*, R_i = R \frac{F(T_i)}{\sum_{j=1}^k F(T_j)}, \theta_i = (\beta R_i)^{\frac{1}{\alpha}} \frac{\Gamma\left(T_i + \beta^{\frac{1}{\alpha-1}}\right)}{\Gamma(T_i) \Gamma\left(1 + \beta^{\frac{1}{\alpha-1}}\right)}$$

ただし, $F(x) := \left(\frac{\Gamma(x + \beta^{\frac{1}{\alpha-1}})}{\Gamma(x)} \right)^{\frac{\alpha}{\alpha-1}}$ であり, $k^*, (T_i^*)_{i=1}^{k^*}$ は最適化問題

$$\max_{k, (T_i)_{i=1}^k} \sum_{i=1}^k F(T_i) \quad \text{s.t. } \sum_{i=1}^k T_i = T \quad \text{の最適解}$$

最適な報酬スケジューリング: 結果

結果: 最適な報酬スケジュールを導出

時間計算量 $O(1)$ で計算可能!

(1) $\beta \geq \beta_0$ のとき, 最適な報酬スケジュールは

$$k = 1, T_1 = T, R_1 = R, \theta_1 = R^{\frac{1}{\alpha}} \left(T - 1 + \beta^{\frac{1}{\alpha-1}} \right)^{\frac{\alpha-1}{\alpha}}$$

動的計画法で
時間計算量
 $O(T^2)$ で計算可能!

(2) $\beta < \beta_0$ のとき, 最適な報酬スケジュールは

$$k = k^*, T_i = T_i^*, R_i = R \frac{F(T_i)}{\sum_{j=1}^k F(T_j)}, \theta_i = (\beta R_i)^{\frac{1}{\alpha}} \frac{\Gamma\left(T_i + \beta^{\frac{1}{\alpha-1}}\right)}{\Gamma(T_i) \Gamma\left(1 + \beta^{\frac{1}{\alpha-1}}\right)}$$

ただし, $F(x) := \left(\frac{\Gamma(x + \beta^{\frac{1}{\alpha-1}})}{\Gamma(x)} \right)^{\frac{\alpha}{\alpha-1}}$ であり, $k^*, (T_i^*)_{i=1}^{k^*}$ は最適化問題

$$\max_{k, (T_i)_{i=1}^k} \sum_{i=1}^k F(T_i) \quad \text{s.t.} \quad \sum_{i=1}^k T_i = T \quad \text{の最適解}$$

最適な報酬スケジューリング: 定性的な結果

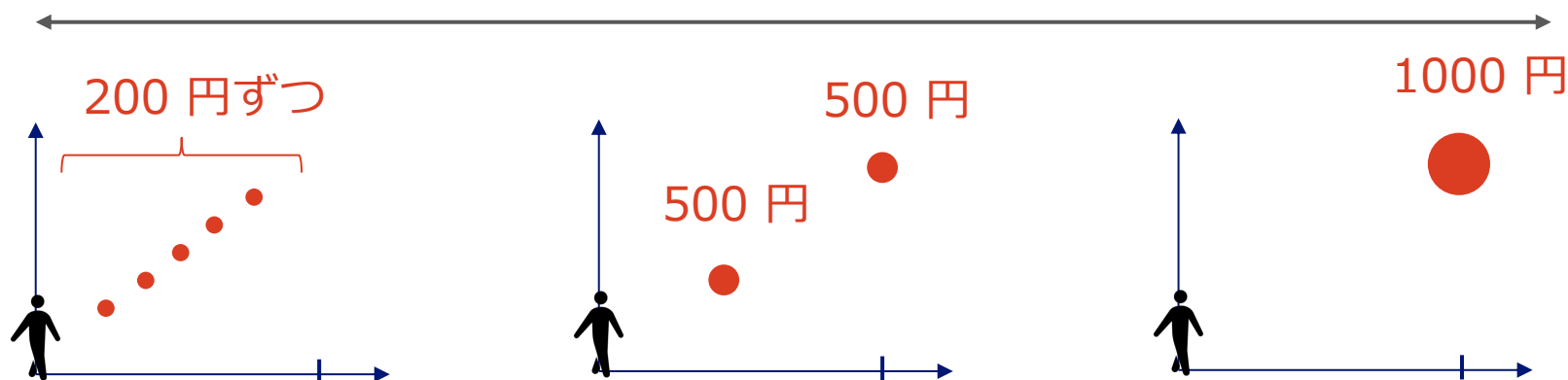


$\alpha = 2$ (二乗コスト) の場合の結果

- $\frac{1}{2} \leq \beta$ の場合, **一括で** 報酬を与えるのが最適
- $\sqrt{2} - 1 = 0.414 \dots < \beta < \frac{1}{2}$ の場合,
 $\min \left\{ T, \left\lfloor \frac{\beta^2}{1-2\beta} \right\rfloor \right\}$ 時間おきに 報酬を与えるのが (概ね) 最適
- $\beta \leq \sqrt{2} - 1 = 0.414 \dots$ の場合, **毎時刻に分割して** 報酬を与えるのが最適

β が小さい (現在バイアス強)

β が大きい (現在バイアス弱)



最適な報酬スケジューリング: 定性的な結果



$\alpha = 2$ (二乗コスト) の場合の結果

- $\frac{1}{2} \leq \beta$ の場合, **一括で** 報酬を与えるのが最適
- $\sqrt{2} - 1 = 0.414 \dots < \beta < \frac{1}{2}$ の場合,
 $\min \left\{ T, \left\lfloor \frac{\beta^2}{1-2\beta} \right\rfloor \right\}$ 時間おきに 報酬を与えるのが (概ね) 最適
- $\beta \leq \sqrt{2} - 1 = 0.414 \dots$ の場合, **毎時刻に分割して** 報酬を与えるのが最適



対象者の現在バイアスの強さに応じて

報酬スケジュールを変更することが重要 であること,

中間的な報酬が重要 な役割を果たすこと,

を数学的に証明

- 現在バイアスの影響を受けた人間の行動を分析するための
新しい数理モデル を提案
- 提案モデルにおいては人間の行動は
数学的に閉形式で書き表すことが可能
- これに基づき,
現在バイアスの **影響の分析・最適な介入策** の導出

今後の課題

- 別のバイアス（サンクコストバイアス, 損失回避など）の考慮
- 最適報酬スケジュール問題において,
複数の報酬を同時に置くことを許す場合の解析

講演者が今まで行ってきた

人間行動モデリング × 数理最適化 に関する研究について紹介

- **前半:**

離散 DC アルゴリズムによる

Collective Graphical Model の近似を用いない MAP 推定

- **後半:**

解析的に扱いやすい

現在バイアスの影響下の人間行動のモデリング手法