一般発表 | Track 3: 情報検索・情報推薦・ソーシャルメディア

2026年3月1日(日) 15:30 ~ 17:40 | G会場

[6G] 推薦システム(最適化/ロバスト性/実運用)

座長:山口 実靖(工学院大学)　コメンテータ:三林 亮太(神戸大学)

15:30 ~ 15:55

[6G-01] LLM推薦におけるユーザ属性を考慮した強化学習型プロンプト最適化

*横田 信徒[1]、張 建偉[1] (1. 岩手大学)

15:55 ~ 16:20

[6G-02] 観測可能な閲覧深度を用いたカルーセルUIのためのランキングバンディット

*安田 琢真[1]、中村 篤祥[2] (1. 北海道大学工学部情報エレクトロニクス学科、2. 北海道大学大学院情報科学研究院)

16:20 ~ 16:45

[6G-03] Robust Recommendation against Shilling Attacks via List Consistency and Counterfactual Neighbor Analysis

*Mo Fan[1,2]、Chen Chongxian[1]、Fan Xin[1]、山名 早人[1] (1. 早稲田大学、2. zozo株式会社)

16:45 ~ 17:10

[6G-04] 小規模言語モデルが抽出する経験価値に基づくゲーム推薦システムの構築

*長尾 羽留[1]、亀谷 由隆[1] (1. 名城大学)

17:10 ~ 17:35

[6G-05] [技術報告] DMM.com における検索・レコメンドの取り組み

*森 雄一郎[1]、田中 久温[1] (1. 合同会社DMM.com)

LLM推薦におけるユーザ属性を考慮した強化学習型プロンプト最適化

横田　信徒†　　張　　建偉†

† 岩手大学理工学部　〒 020-8551 岩手県盛岡市上田 4-3-5
E-mail: †{s0622060,zhang}@iwate-u.ac.jp

あらまし　近年，大規模言語モデル（LLM）の高い推論能力を推薦システムに応用する研究が注目されている．しかし，すべてのユーザに対して固定のテンプレートを使用する従来のプロンプト手法では，個々のユーザの嗜好や置かれた状況を十分に反映できないという課題がある．この問題を解決するため，本研究では強化学習を用いてユーザごとに最適なプロンプト構成要素を自動選択する手法「RPP」を拡張した新たな枠組みを提案する．具体的には，ユーザ毎の過去の Rating に基づいたベクトルを強化学習エージェントの入力に組み込むことで，ユーザ毎の属性を考慮したプロンプト生成を実現する．さらに，「時間帯」に基づいて各ユーザへの推薦を動的に変化させる行動空間を追加実装した．3 つの公開データセットおよび 2 種類の LLM を用いた評価実験の結果，提案手法はベースラインと比較して多くの条件で推薦精度（NDCG）を向上させ，特に NDCG@1 においては最大で 2.88%の改善率を記録した．本結果は，ユーザの評価傾向と時間的文脈を統合した動的なプロンプト生成が，推薦精度の向上に有効であることを示している．

キーワード　推薦システム，LLM，強化学習

1 はじめに

近年，大規模言語モデル（LLM）の高い推論能力や言語理解能力を推薦システムに応用する研究が注目されている．ユーザに対してパーソナライズされた応答や推薦を行うために，モデルにユーザの行動履歴などの外部知識を取り込む研究や，プロンプトエンジニアリングによる精度向上の研究が進められている．通常，推薦タスクにおいては，ユーザの嗜好や置かれた状況に合わせて対話や提示内容を変化させる必要があるが，従来の LLM ベースの推薦モデルでは，すべてのユーザに対して固定のテンプレートを使用する「Task-wise prompting」が主流であり，個々のユーザへの適応には限界があるとされている [1] [2].

これに対し，Mao らは強化学習（RL）を用いてユーザごとに最適なプロンプト構成要素を自動選択する手法，Reinforced Prompt Personalization (RPP) を提案した [3]．この手法は，プロンプトを「役割の設定」や「履歴の長さ」などの構成要素に分解し，強化学習エージェントがユーザのインタラクション履歴に基づいて最適な組み合わせを探索することで，推薦精度の向上を実現している．しかし，このモデルは依然として，ユーザ状態の解釈において改善の余地を残している．第一に，RPP は履歴系列をテキストとして扱うことで意味理解を促しているものの，各アイテムに対するユーザの具体的な評価値（Rating）等の情報は，プロンプト生成の過程で十分に活用されていない．第二に，「時間的文脈（Timestamp）」の欠如である．ユーザの行動は時間帯によって動的に変化するもの (朝の通勤時間や夜の余暇時間等) であり，従来の RPP の行動空間ではこうした時間経過やタイミングによる嗜好の変化を組み込めていない．

そこで，本研究では，RPP のフレームワークを拡張し，より詳細なユーザコンテキストを反映可能な新たな枠組みを提案する．具体的には，ユーザごとの過去の Rating に基づいたベクトルを強化学習エージェントの状態入力（State）に組み込むことで，ユーザ固有の評価傾向を考慮したプロンプト生成を実現する．さらに，「Timestamp（時間帯）」に基づき各ユーザへの推薦アプローチを動的に変化させるよう行動空間（Action Space）を拡張する．

本実験では，提案手法を用いた場合，ベースラインと比較してより高精度な推薦が可能であるか検証することを目的とする．実験にはベースのモデルとして Mao らの RPP を採用する [3]．複数の公開データセットを用いて，Rating に基づいたベクトルと Timestamp を導入した拡張モデルの学習および推論を行い，NDCG 等の評価指標を用いてその有効性を評価する．具体的には，GPT-4o-mini および Llama-3.1-8B を用いた実験において，提案手法は多くの条件下でベースラインを上回る精度を達成した．特に，NDCG@1 において顕著な向上が確認され，ML-1M データセットでは最大 2.88%の改善率を記録した．これらの結果から，Rating 情報によるユーザ状態の精緻化と，時間帯に基づくプロンプト構成要素の動的な選択が，有効であることが示された．

2 関連研究

2.1 LLM を用いた推薦システム

近年，大規模言語モデル（LLM）の強力な意味理解能力と推論能力を推薦システム（RS）に活用する研究が盛んに行われている．初期の研究では，LLM を推薦タスクに適応させるためにファインチューニングを行う手法が提案された．しかし，これらは計算コストが高く，頻繁に更新が必要な推薦システムの運用には課題が残る．これに対し，モデルのパラメータを更新せず，入力テキスト（プロンプト）の工夫によって性能を引き出す「プロンプトエンジニアリング」が注目されている．既存の多くのアプローチは「Task-wise prompting」と呼ばれ，特定の

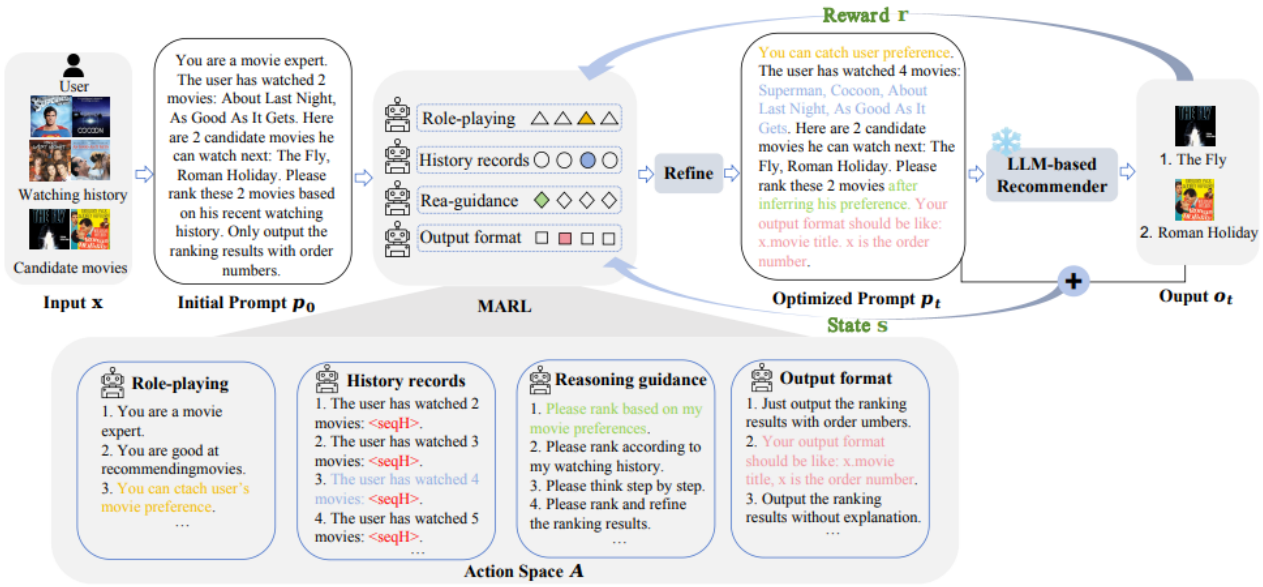

図 1　RPP の概要 [3]

タスクに対して全ユーザ共通の固定テンプレートを使用する．しかし，この方法は実装が容易である反面，ユーザごとの多様な嗜好や文脈を捉えきれないという問題点が指摘されている．

2.2　プロンプトエンジニアリングと動的最適化

Task-wise prompting の限界を克服するため，個々の入力インスタンスに応じてプロンプトを自動的に変化させる手法が提案されている [4] [5]．このアプローチは，全てのユーザに固定のテンプレートを適用するのではなく，ユーザごとの文脈に合わせてプロンプトを個別化することを目指すものである．近年では，このプロンプト設計の自動化という部分において，強化学習（RL）を導入する研究が進められている [3] [5]．これらの手法は，プロンプトを構成する要素（役割，履歴の長さ，推論ガイダンスなど）を探索空間と見なし，ユーザのインタラクション履歴に基づいて最適な構成要素を動的に選択する．これにより，従来の人手による設計やヒューリスティックな探索に依存せず，各ユーザの嗜好や状態に適応したパーソナライズされたプロンプト生成が可能となる．

2.3　コンテキスト認識型推薦

推薦システムにおいて，ユーザの行動履歴（アイテム ID の列）だけでなく，その行動が発生した「文脈（コンテキスト）」を考慮することは，推薦精度を向上させる上で重要なアプローチである．従来の推薦モデル，例えば SASRec [6] などのシーケンシャル推薦では，インタラクションの時刻（Timestamp）を埋め込みベクトルとしてモデルに組み込むことで，ユーザの嗜好の時間的な変化や，特定の時間帯における周期的な行動パターンを捉えている．また，Rating 情報（評価値）は，ユーザがアイテムに対して抱いた嗜好の強さを表す明示的なフィードバック（Explicit Feedback）である．単なるクリック履歴（Implicit Feedback）とは異なり，Rating を利用することで，ユーザが真に満足したアイテムとそうでないアイテムを区別して学習することが可能となる．これらのようなコンテキスト情報の活用は，従来のニューラル推薦モデルにおいて標準的に行われており，その有効性が広く確認されている．

3　先 行 研 究

3.1　RPP

Mao らは，推薦タスクにおけるプロンプト最適化をマルコフ決定過程 (MDP) として定式化し，多エージェント強化学習 (MARL) を用いてユーザごとに最適なプロンプト構成要素を選択する手法 RPP を提案した．ここで図 1 は RPP の全体アーキテクチャを示している．

3.1.1　定　式　化

ユーザ u の対話履歴（インタラクション履歴）は $H_u = \{i_1, i_2, ..., i_n\}$ で表される．ここで i_k はユーザが過去に接したアイテムを表し，n は履歴の長さである．また，推薦候補となるアイテム集合を $C = \{c_1, ..., c_M\}$ とし，M は候補映画数であり，本研究では $M = 10$ に設定している．この 2 つをモデルへの入力 x とする．RPP の目的は，以下の式 3.1 のように，LLM の推薦結果 y_{LLM} の精度（報酬 R）を最大化する最適なプロンプト p^* を探索することである．

$$p^* = \underset{p \in \mathcal{P}}{\operatorname{argmax}}\, R(y_{\text{LLM}}(p, x)) \tag{3.1}$$

ここで $\mathcal{P}$ は可能なプロンプトの空間を表す．

3.1.2　State Space

強化学習エージェントが観測する状態 S は，環境の十分な情報を含む必要がある．状態の定義は初期状態と更新状態で異なる．まず，初期状態 s_0 は，従来の推薦モデル（LightGCN）から得られるユーザ埋め込み $\mathbf{u}$ を用いて式 3.2 のように初期化さ

れる．但し，u の次元数は 64 次元である．

$$s_0 = u^{(lightGCN)} \tag{3.2}$$

次に，ステップ t における状態 s_t は，以下の式 3.3 のように，現在のプロンプト p_t と LLM の出力結果 o_t に基づいて更新される．具体的には，BERT を用いてエンコードされたプロンプト表現 $e_t^{(p)}$ と，GRU を用いてエンコードされた推薦結果 $e_t^{(o)}$ の和として定義される．

$$s_t = e_t^{(p)} + e_t^{(o)} = \mathrm{BERT}(p_t) + \mathrm{GRU}(\hat{i}_1, ..., \hat{i}_M) \tag{3.3}$$

ここで $\hat{i}_j$ は LLM が出力したランキングにおける j 番目のアイテムを表す．また，ここでの s_t における次元数も 64 次元として定義される．

3.1.3 Action Space

探索効率とプロンプト品質のバランスを保つため，RPP は文レベルの候補セットを行動空間 $\mathcal{A}$ として定義する．具体的には，以下の 4 つのパターンを最適化の対象とする．Role-playing: LLM に特定の役割（映画の専門家など）を付与する [7] [8]. History records: 履歴情報の系列長を調整し，短期・長期の興味に対応させる [9]．Reasoning guidance: 推論プロセス（CoT [10] や Refinement [11] など）を指示する．Output format: 推薦結果の出力形式を指定する [12]．各エージェント k は，それぞれの候補文集合 $\mathcal{A}_k$ から最適な行動 $a_t^{(k)}$ を選択する．

3.1.4 最　適　化

各パターン k を個別に最適化するため，Centralized Training with Decentralized Execution (CTDE) パラダイム [13] に基づく Actor-Critic アーキテクチャ [14] を採用している．各エージェントは Actor $g^{(k)}$ (式 3.4) と Critic $h^{(k)}$ (式 3.5) を持つ．

$$g^{(k)}(s_t) = a_t^{(k)}, prob_t^{(k)} \tag{3.4}$$

$$h^{(k)}(s_t) = v_t^{(k)} \tag{3.5}$$

として定義される．ここで $v_t^{(k)}$ は Critic が推定した状態価値，$\mathrm{prob}_t^{(k)}$ は Actor が出力した行動選択確率である．また，報酬 r_t(式 3.6) にはランキング評価指標である NDCG@10 を用いる．

$$r_t = \mathrm{NDCG@10}(o_t) \tag{3.6}$$

学習においては，将来の報酬の累積和 $\hat{R}_t = r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n v_{t+n}^{(k)}$ を最大化するようにパラメータが更新される．ここでの γ は，長期報酬にも重みをおきつつ短期報酬を重視するための割引率を示している．Critic の損失関数 $L_c^{(k)}$ および Actor の損失関数 $L_a^{(k)}$ は以下の式 3.7，3.8 で定義される．ここで，N はバッチサイズを表す．

$$L_c^{(k)} = \frac{1}{N} \sum (\hat{R}_{t-1} - v_t^{(k)}) \tag{3.7}$$

$$L_a^{(k)} = \frac{1}{N} \sum \log(\mathrm{prob}_t^{(k)} (\hat{R}_{t-1} - v_t^{(k)})) \tag{3.8}$$

3.1.5 最 終 目 的

最終的に，RPP は $K = 4$ のエージェントが協力して一つの最適なプロンプトを生成する多エージェントシステムとして動作する．最終的な目的は先述の通り，LLM の推薦結果 y_{LLM} の精度（報酬 R）を最大化する最適なプロンプト p^* を探索することであり，ランキング出力の質を高めることである．

3.2 DICE

既存の推薦モデルの多くは，ユーザのクリックや視聴などの行動をそのまま「ユーザの興味」として学習する．しかし Zheng らは，現実の行動は，ユーザ自身の純粋な興味 (Interest) と，人気度などの社会的要因による同調 (Conformity) の双方によって引き起こされるものだと主張し，因果推論の枠組みを取り入れ，ユーザの行動要因を「興味」と「同調」に区別する手法 Disentangling Interest and Conformity(DICE) [15] を提案した．

3.2.1 ベクトルの定義

ユーザの行動要因を明確に分離するため，DICE は従来のモデルとは異なり，ユーザ u とアイテム i に対して，それぞれ「興味（Interest）」と「同調（Conformity）」に対応する独立した埋め込みベクトルを割り当てる．まず，ユーザの本質的な嗜好とアイテムの属性特性を表すベクトルを以下の式 3.9 のように定義する．

$$\mathbf{u}^{(\mathrm{int})}, \mathbf{i}^{(\mathrm{int})} \in \mathbb{R}^d \tag{3.9}$$

次に，ユーザの同調しやすさとアイテムの人気度（トレンド性）を表すベクトルを以下の式 3.10 のように定義する．

$$\mathbf{u}^{(\mathrm{con})}, \mathbf{i}^{(\mathrm{con})} \in \mathbb{R}^d \tag{3.10}$$

但し，上式における d は次元数を表し，本研究では $d = 64$ としている．これらを用いることで，あるユーザ u がアイテム i に対して持つ「興味スコア $S_{ui}^{(\mathrm{int})}$」と「同調スコア $S_{ui}^{(\mathrm{con})}$」を計算する．(式 3.11)

$$S_{ui}^{(\mathrm{int})} = \mathbf{u}^{(\mathrm{int})\top} \mathbf{i}^{(\mathrm{int})}, \quad S_{ui}^{(\mathrm{con})} = \mathbf{u}^{(\mathrm{con})\top} \mathbf{i}^{(\mathrm{con})} \tag{3.11}$$

3.2.2 データ分割と定式化

DICE では，推薦システムにおけるデータ生成プロセスを構造的因果モデル（SCM）として定式化する．具体的には，前節で定義した各スコアを用いることで，最終的なインタラクションスコア S_{ui} (式 3.12) は「本質的な興味」と「同調性」の和として決定されると定義する．

$$S_{ui} = \underbrace{\mathbf{u}^{(\mathrm{int})\top} \mathbf{i}^{(\mathrm{int})}}_{S_{ui}^{(\mathrm{int})}} + \underbrace{\mathbf{u}^{(\mathrm{con})\top} \mathbf{i}^{(\mathrm{con})}}_{S_{ui}^{(\mathrm{con})}} \tag{3.12}$$

この構造において重要な点は，アイテムの人気度が高い（$\mathbf{i}^{(\mathrm{con})}$ が大きい）場合，興味スコア $S_{ui}^{(\mathrm{int})}$ が低くても，同調スコア $S_{ui}^{(\mathrm{con})}$ が高くなることで，結果としてクリック行動が観測され得るという点である．

この交絡を解き，真の興味 $\mathbf{u}^{(\mathrm{int})}$ を識別するために，DICE はアイテムの人気度に基づいたデータ分割を行う．まず，学習データの構築にあたり，評価値（Rating）に基づくインタラク

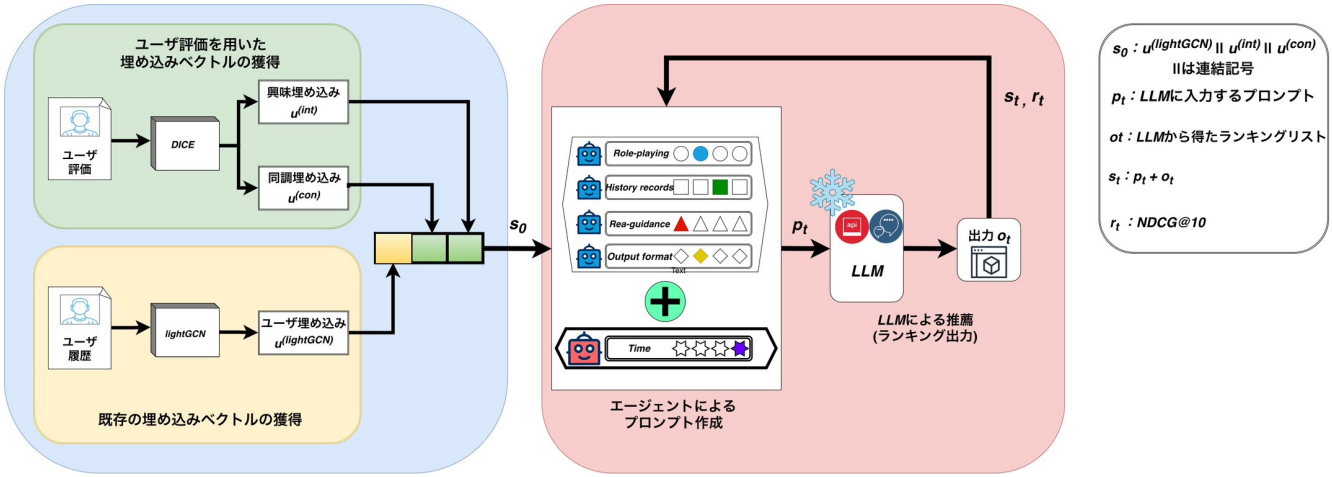

図 2　提案手法の概要

ションの定義を行う．具体的には，Rating が 4 以上のインタラクションを「ポジティブ（正例）」，それ以外（Rating 1〜3 のアイテムまたは未観測のアイテム）を「ネガティブ（負例）」と定義した．また，正例からなる集合をポジティブ集合，負例からなる集合をネガティブ集合と定義する．

次に，ユーザ u と正例 i と負例 j からなる組 $\langle u, i, j\rangle$ の構築を行う．正例 i は，ユーザ u のポジティブ集合からランダムに選択される．一方で，負例 j の選択には，DICE で提案された Popularity based Negative Sampling with Margin (PNSM) を採用する．これは，カリキュラム学習 [16] に触発され提案されたものであり，単純なランダムサンプリングではなく，正例の人気度 pop_i に対して一定のマージンを持つ人気度 pop_j のアイテムを，ネガティブ集合から選出する手法である．これにより，人気度の高低差を強調し，因果関係の分離を容易にする．ここで pop_i とは，データセット o 内のアイテム i における，インタラクションの総数で算出される．

最終的に DICE は，この $\langle u, i, j\rangle$ 内のアイテム i, j の人気度 pop_i, pop_j の大小関係に基づき，学習用データセット o_{train} を以下の 2 つのサブセット O_1, O_2 に分割して学習を行う．

- **dataset O_1 (Positive item is more popular)**
  $pop_i > pop_j$ であるデータセット．ユーザは人気のあるアイテム i を選択しているため，この行動には「興味」と「同調」の両方が寄与している可能性がある．ここでは同調の埋め込み $\mathbf{u}^{(con)}$ を学習させつつ，興味埋め込み $\mathbf{u}^{(int)}$ の学習も行う．
- **dataset O_2 (Negative item is more popular)**
  $pop_j > pop_i$ であるデータセット．ユーザは人気のあるアイテム j よりも，人気のないアイテム i をあえて選択しているため，この行動は「同調」ではなく「強い興味」によって引き起こされたと推測できる．したがって，ここでは興味の埋め込み $\mathbf{u}^{(int)}$ を重点的に学習させる．

DICE は，この O_1 と O_2 それぞれに対して異なる損失関数を適用するマルチタスク学習を行うことで，単一のインタラクション履歴から興味と同調の分離を実現する．

3.2.3　最　適　化

興味と同調を明確に分離して学習するために，DICE はトレーニングデータを「興味によるクリック」と「同調によるクリック」に分類し，それぞれに特化した損失関数を適用する．具体的には，全体としてのクリック予測損失 $L_{(\text{click})}^{o_1+o_2}$(式 3.13，3.14，3.15，3.16) は以下として定義する．なお，上付き文字 t は Total を意味し，興味と同調の埋め込みベクトルを連結（Concatenation）した，ユーザおよびアイテムの包括的な表現ベクトルを表す．

$$L_{(\text{click})} = \sum_{(u,i,j)\in o} \text{BPR}(\langle u^t, i^t\rangle, \langle u^t, j^t\rangle) \tag{3.13}$$

$$u^t = u^{(\text{int})} \,\|\, u^{(\text{con})} \tag{3.14}$$

$$i^t = i^{(\text{int})} \,\|\, i^{(\text{con})} \tag{3.15}$$

$$j^t = j^{(\text{int})} \,\|\, j^{(\text{con})} \tag{3.16}$$

ここで，損失関数で用いられている BPR は，Bayesian Personalized Ranking [17] においてペアワイズな仮定．つまり，ユーザが低評価・未観測なアイテムは，ユーザがこう評価したアイテムに比べて関心が低いという仮定に基づいた損失関数であり，ポジティブアイテムのスコアがネガティブアイテムのスコアが高いという大小関係が成立する確率を最大化するように学習が行われる．

また，人気度バイアスの影響を受けにくい学習データセットを用いて興味の埋め込みを最適化する Interest Loss ($L_{(\text{int})}$) (式 3.17) と，人気度に基づいて同調の埋め込みを最適化する Conformity Loss ($L_{(\text{con})}$) 式 [3.18，3.19，3.20] を以下として定義する．

$$L_{(\text{int})} = \sum_{(u,i,j)\in o_2} \text{BPR}\left(S_{ui}^{(\text{int})}, S_{uj}^{(\text{int})}\right) \tag{3.17}$$

$$L_{(\text{con})} = L_{(\text{con})}^{o_1} + L_{(\text{con})}^{o_2} \tag{3.18}$$

$$L_{(\text{con})}^{o_1} = \sum_{(u,i,j)\in o_1} \text{BPR}\left(S_{ui}^{(\text{con})}, S_{uj}^{(\text{con})}\right) \tag{3.19}$$

$$L_{(\text{con})}^{o_2} = \sum_{(u,i,j)\in o_2} -\text{BPR}\left(S_{ui}^{(\text{con})}, S_{uj}^{(\text{con})}\right) \tag{3.20}$$

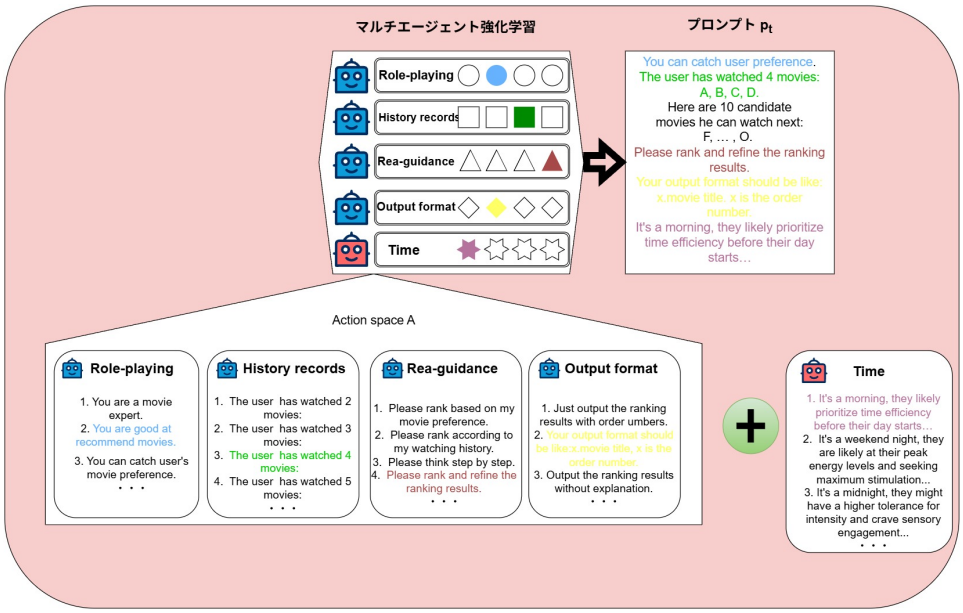

図 3　時間的文脈に基づく行動空間の拡張

さらに，獲得される 2 つのベクトル $\mathbf{u}^{(\mathrm{int})}$ と $\mathbf{u}^{(\mathrm{con})}$ が互いに独立した情報を保持し，冗長にならないようにするために，Discrepancy Loss ($L_{(\mathrm{dis})}$) を導入する．これは 2 つのベクトルの相関を最小化する制約項として機能する．

3.2.4　最 終 目 的

最終的な目的関数 L_{DICE} は式 3.21 のように，クリック予測損失 $L_{(\mathrm{click})}$ に，これら 3 つの損失を加えた加重和として定義される．

$$L_{\mathrm{DICE}} = L_{(\mathrm{click})} + \alpha(L_{(\mathrm{int})} + L_{(\mathrm{con})}) + \beta L_{(\mathrm{dis})} \tag{3.21}$$

ここで α,β はハイパーパラメータであり，初期設定として $\alpha = 0.1$，$\beta = 0.01$ で設定した．なお α に関しては，先述の PNSM に基づき各エポック終了後に 0.9 倍ずつ減衰させた．これは，PNSM により学習初期にマージンが大きいペアが選ばれ，学習が進むにつれてマージンが小さいペアになっていき次第に「興味」「同調」の分離の信頼性が下がってしまうための設定である．

4　提 案 手 法

提案手法の目的は，ユーザの潜在的な評価傾向（興味と同調性）および時間的文脈を考慮したプロンプトを生成し，推薦精度を向上させることである．図 2 に提案手法の概要を示す．

先行研究の RPP は，ユーザ ID の埋め込みを初期状態として利用し，4 つの固定された行動パターンのみを扱っていた．これに対し提案手法では，DICE によって分離されたベクトルを初期状態に統合し，さらに時間帯（Timestamp）に基づく新たな行動エージェントを追加する．これが先行研究と本研究の提案手法との主な違いである．

4.1　DICE ベクトルの統合

ユーザの状態をより詳細に表現するために，DICE [15] を用いて事前学習された「興味埋め込み $\mathbf{u}^{(\mathrm{int})}$」と「同調埋め込み $\mathbf{u}^{(\mathrm{con})}$」を利用する．従来の RPP では，初期状態 s_0 は単純な lightGCN によるユーザ埋め込みで初期化されていたが，提案手法では以下の手順で初期状態を構築する (図 2).

まず，事前学習済みの DICE モデルから得られた 2 つのベクトル $\mathbf{u}^{(\mathrm{int})}, \mathbf{u}^{(\mathrm{con})} \in \mathbb{R}^d$ を以下の式 4.1 のように RPP の状態に連結（Concatenation）する．

$$s_0 = u^{(\mathrm{lightGCN})} \,\|\, u^{(\mathrm{int})} \,\|\, u^{(\mathrm{con})} \tag{4.1}$$

次に，強化学習エージェントの状態空間の次元に合わせる．この処理により，エージェントはユーザが「純粋な興味で動くタイプ」か「トレンドに同調しやすいタイプ」かといった傾向を，数値的な特徴量として観測した状態でプロンプト探索を開始することが可能となる．

4.2　時間的文脈に基づく行動空間の拡張

従来の RPP の行動空間と，提案手法の行動空間ではその構成要素に違いがある．RPP では，「役割設定」，「履歴長」，「推論」，「出力形式」の 4 つのエージェントが協力してプロンプトを作成する．しかし，これらは固定的な設定であり，ユーザがアクセスした瞬間の状況（時間帯など）は考慮されていない．そこで提案手法では，5 つ目のエージェントとして「時間」を導入する．図 3 に概要を示す．

このエージェントは，行動が発生した時刻 t を入力とし，時間帯（朝，昼，夕方，深夜等）に応じた適切な指示文を選択する．例えば，深夜帯（Mid Night）のアクセスであれば，次のような指示文が候補となる．"It's a Mid Night, they might have a higher tolerance for intensity and crave sensory engagement. Recommend a gripping or atmospheric movie that offers a deep sense of immersion."

最終的なプロンプト p は，式 4.2 のように既存の 4 つの構成要素 $a^{(1\sim4)}$ に，時間的文脈の構成要素 $a^{(5)}$ を加えた 5 つの文を

表 1　データセット設定

Dataset	Users	Items	Interactions	Density
MovieLens-1M	6,040	3,885	1,000,210	4.26%
Amazon Games	50,545	16,858	389,718	0.04%
Yelp	31,668	38,048	1,561,406	0.13%

表 2　ハイパーパラメータの設定

Parameter	Value
割引率 (γ)	0.95
学習率	$1e^{-4}$
Optimizer	Adam
temperature	0.2
DICE ベクトルの次元数	128

つなげたものとして生成される．

$$p = [a^{(1)}, a^{(2)}, a^{(3)}, a^{(4)}, a^{(5)}] \tag{4.2}$$

これにより，ユーザの長期的な嗜好（DICE ベクトル）と一時的な状況（Timestamp）の両方を反映した柔軟な推薦が可能となる．

5 実　　験

5.1 実験設定

5.1.1 データセット

実験には，表 1 に示すように，推薦システムの評価で広く用いられている 3 つの公開ベンチマークデータセット，MovieLens-1M(ML-1M) [18]，Amazon Games [19]，Yelp [20] を使用する．ML-1M は映画に対する評価データ，Amazon Games はゲーム製品のレビューデータ，Yelp は店舗に対するチェックインやレビューのデータである．いずれのデータセットも，ユーザ ID，アイテム ID，評価値（Rating），タイムスタンプを含んでいる．データの前処理として，データの信頼性を保つため，インタラクション数が 5 未満のユーザおよびアイテムを除外する（5-core filtering）．

データの分割には，タイムスタンプに基づく Leave-one-out 方式を採用した．各ユーザの履歴を時間の昇順に並べ替え，最も新しい 1 件をテストデータ，その直前の 1 件を検証データ，それ以前を学習データとした．なお，n をユーザの履歴長とした場合の学習データ（$n-2$ 時点までの履歴）内に正例が 1 件も存在しない場合はモデルの学習が困難であるため，少なくとも 1 件以上の正例を有するユーザのみを抽出して利用した．

また，RPP [3] の設定に準拠してユーザのサンプリングを行った．具体的には，全ユーザの中からランダムに抽出した 200 名を強化学習エージェントの学習用ユーザ，それらとは重複しない別の 100 名を最終評価用のテストユーザとして設定した．

5.1.2 モデルと実験詳細

バックボーンとなる大規模言語モデル（LLM）には，OpenAI が提供する GPT-4o-mini（API 利用）と，Meta 社の Llama-3.1-8B（ローカル環境）の 2 種類を採用した．

- **GPT-4o-mini**
  クローズドソースモデルの代表．世間で多く利用されており，圧倒的な推論能力と知識量を持つ商用モデル．高い性能を出すことが期待される．
- **Llama-3.1-8b**
  オープンソースモデルの代表．パラメータ数が比較的少ない (8B) モデルであり，ローカル環境で動作可能．コストやプライバシーの観点で商用 API を利用できない環境でも利用できる．

これらの LLM により，クローズドソースとオープンソースの双方のモデルにおける提案手法の有効性を検証する．初期状態の構築には，各データセットで事前に学習させた DICE [15] の出力ベクトルを利用する．強化学習のハイパーパラメータは，先行研究 RPP に基づいて設定を行った．主な設定値を表 2 に示す．割引率 γ は 0.95 とし，Actor と Critic の学習率には $1e^{-4}$ を設定した．最適化手法には Adam を利用した．LLM のランダム性を減らすため，temperature は 0.2 に設定した．これが低いほど，正確で一貫性のある出力がされやすくなる．サーバは NVIDIA T4 GPU を利用した．

また，本研究の候補アイテム数 M は 10 に設定し，これは訓練データからランダムに選択され，テストデータから得られた正解アイテムを含んでいる．行動空間における「Role-playing」「Rea-guidance」「Output format」「Time」のパターンをそれぞれ 3，9，5，7 個の選択肢とした．

5.2 評価指標 (Metrics)

本研究では，トップ K 推薦（Top-K Recommendation）のタスクにおいて，提案手法が RPP に比べてユーザの嗜好を正しく予測できているかを評価する．モデルが生成した推薦リストの精度を測るため，評価指標 NDCG@K(Normalized Discounted Cumulative Gain) [21] を利用する．但し，K = 1，5，10 とする．これは，正解アイテムが推薦リストの上位にあるほど高い値となる指標であり，ランキングの質を評価する．

NDCG は，以下の式 5.1 で定義される．

$$\mathrm{NDCG@}K = \frac{\mathrm{DCG@}K}{\mathrm{IDCG@}K} \tag{5.1}$$

また，DCG@k は以下の式 5.2 定義される．

表 3　GPT-4o-mini における推薦精度

	ML-1M			Games			Yelp		
NDCG@K	K=1	K=5	K=10	K=1	K=5	K=10	K=1	K=5	K=10
RPP(baseline)	0.347	0.587	0.651	0.510	**0.668**	0.727	0.485	0.706	0.748
Ours	**0.357**	**0.591**	**0.656**	**0.524**	0.661	**0.738**	**0.490**	**0.708**	**0.761**
改善率	**+2.88 %**	**+0.68 %**	**+0.77 %**	**+2.74 %**	-1.06 %	**+1.51 %**	**+1.03 %**	**+0.28 %**	**+1.73 %**

表 4　Llama-3.1-8B における推薦精度

	ML-1M			Games			Yelp		
NDCG@K	K=1	K=5	K=10	K=1	K=5	K=10	K=1	K=5	K=10
RPP(baseline)	0.811	**0.850**	0.862	0.782	**0.822**	0.859	0.814	0.860	0.886
Ours	**0.827**	0.846	**0.871**	**0.791**	0.820	**0.864**	**0.826**	0.860	**0.889**
改善率	**+1.97 %**	-0.47 %	**+1.04 %**	**+1.15 %**	-0.24 %	**+0.58 %**	**+1.47 %**	0.00 %	**+0.34 %**

表 5　Ablation Study の結果 (ML-1M)

Metrics	**NDCG@1**	**NDCG@5**	**NDCG@10**
Ours	**0.357**	0.591	**0.656**
w/o DICE	0.347	**0.608**	0.653
w/o Timestamp	0.355	0.590	0.651
RPP	0.347	0.587	0.651

$$\mathrm{DCG@}K = \sum_{i=1}^{K} \frac{2^{\mathrm{rel}_i} - 1}{\log_2(i+1)} \tag{5.2}$$

ここで，rel_i はランキング順位 i における関連性スコアを表す．分母の $i+1$ の対数は，順位に基づく減衰を表す．また，IDCG@K は理想的なランキングの IDCG@K は理想的なランキングの DCG@K だが，本タスクにおいて正解アイテムは常に 1 つであり，IDCG@K = 1 であるため，DCG@K を求めるのは，NDCG@K を求めたことと同義である．

6　結果と考察

6.1　推薦精度の評価

本研究では，3 つのデータセットにおいて提案手法 (Ours) とベースライン手法（RPP）の推薦精度を比較した．実験結果を表 3，4 に示す．太字は，提案手法とベースラインを比べた結果，より精度が優れている手法を示しており，下線部は，既存手法に比べて 1.0 %以上の改善率を記録した結果を示している．

実験の結果，全体の傾向として提案手法は，ほとんどの条件においてベースラインを上回る，もしくは同等の精度を達成した．特に，GPT-4o-mini を用いた ML-1M データセットにおいては，全ての指標でベースラインを凌駕していた．また，NDCG @ 1 においては顕著な性能向上が見られ，表に示す通り全てのデータセットにおいて 1.0 %を超える改善率を記録した．以上より，提案手法の有効性が示されたと言える．

6.1.1　データセットの特性による影響

データセットごとの結果詳細に着目すると，ML-1M や Yelp と比較して，Amazon Games における改善率は限定的，あるいは一部の指標（GPT-4o-mini における NDCG@5 など）で低下が見られた．表 1 に示す通り，ML-1M と Yelp の密度（Density）がそれぞれ 4.26%，0.13%であるのに対し，Amazon Games は 0.04%と極めて疎なデータセットである．提案手法の核となる DICE は，インタラクションデータから「興味」と「同調」を分離するために十分な学習データを必要とする．データが過度に疎である場合，人気度の偏りやユーザの行動パターンの学習が不安定になり，分離精度の低下を招いた可能性が高い．このことから，提案手法は一定以上のインタラクション密度を持つドメインにおいて，特に高い効果を発揮すると考えられる．

6.1.2　LLM のモデル能力による差異

モデル間の比較を行うと，ベースライン（RPP）のスコア絶対値は Llama-3.1-8B の方が高い傾向にあるが，提案手法による「改善幅」は GPT-4o-mini の方が大きい結果となった．これは，GPT-4o-mini の方がプロンプト内の複雑な指示（時間的文脈のニュアンス等）をより柔軟に解釈し，ランキング生成に反映させる能力が高かったためと推察される．一方，Llama-3.1-8B は元々の推薦能力が高く，プロンプトの微細な変更による感度が相対的に低かった可能性がある．

6.2　Ablation study

提案手法の各構成要素が精度向上にどの程度寄与しているかを検証するため，各要素を除外したモデルとの比較を行う．実験結果を表 5 にまとめる．太字は，同一の評価指標において，最も精度が高いことを示している．ここで，各要素毎に実験結果をまとめた．

- **DICE ベクトルの統合の効果 (w/o Timestamp)**
  時間的文脈を除外したモデルは，ベースラインと比較して特に NDCG@1 において明確な向上が見られた．一方で，NDCG@10 は精度が向上しないという現象が見られた．この結果は，DICE ベクトルの統合によって，ユーザの真の興味に合致するアイテムを特定する能力が向上したことを示している．つまり，ランキング全体の網羅性よりも，最上位に正解をランク付けする「ピンポイントな精度)」が強化されたと言える．

- **時間的文脈に基づく行動空間の拡張の効果 (w/o DICE)**
  DICE を除外したモデル（時間的文脈のみ追加）は，ベースラインと比較して NDCG@5 で全モデルの中で最高の精度を記録した．NDCG@10 に関しても僅かながらベースラインから精度が向上していた．これは，時間帯という「状況」の情報が加わることで，そのタイミングでユーザが選びそうな候補を幅広く拾えるようになったためと考えられる．DICE のように，ユーザの真の興味を特定する能力が提案手法に比べて低いため，一般的な人気アイテムも候補に残りやすく，結果としてランキング全体（@5, @10）の質が底上げされたと言える．

6.2.1 NDCG@1 の向上と NDCG@5 の低下に関する考察

実験結果において特筆すべき点は，提案手法が NDCG@1 を大幅に向上させた一方で，NDCG@5 においては既存手法を下回るケース（逆転現象）が見られたことである．これは，DICE の導入によりニッチなアイテムへの推薦能力が向上した反面，一般的な人気アイテムに対する選好が過小評価されたことに起因すると考えられる．結果として，ユーザの多様な嗜好性に対し，モデルが過度に「ユーザの興味」のみにフォーカスしてしまった可能性が示唆される．

7 まとめと今後の展望

7.1 ま と め

本研究では，大規模言語モデル（LLM）を用いた推薦システムにおけるプロンプト最適化手法「RPP」を拡張し，ユーザ固有の評価傾向と時間的文脈を統合する新たなフレームワークを提案した．具体的には，DICE モデルを導入してユーザの行動要因を「真の興味」と「同調」に分離し，さらに行動空間に「Timestamp（時間帯）」を追加することで，動的なプロンプトの生成を実現した．

3 つの公開データセットおよび 2 種類の LLM を用いた評価実験の結果，提案手法はベースラインである RPP と比較して，多くの条件で高い推薦精度を達成した．特に，NDCG@1 においては，最大+2.88 %の改善を記録するなど，既存手法を上回る性能を示した．一方で，データセットのスパース性が高い場合や，評価指標の K が大きい場合（NDCG@5 など）には改善が限定的となる課題も明らかになった．

7.2 今後の展望

本研究を発展させるための今後の課題として，以下の点が挙げられる．

- **人気アイテムの適切な扱い**
  提案手法では，DICE の導入により，NDCG@5 においてスコアが低下するという逆転現象が見られた．これは実際には，ニッチなアイテムが好きなユーザが人気アイテムを好むケースも存在する．したがって，単にニッチなアイテムを推薦するだけでなく，ユーザが「あえて流行を求めているタイミング」を強化学習エージェントが学習するなどのように，人気アイテムの推薦とニッチなアイテムの推薦を動的に切り替えるような，より柔軟な制御機構の検討が必要である．
- **データセットによらない頑健なシステムの構築**
  本実験において，ML-1M や Yelp では精度向上が確認された一方で，Amazon Games データセットでは一部の指標で改善が限定的であった．これはデータのスパース性に起因すると考えられるため，データが疎な環境下でも DICE の分離学習を安定させるための正則化手法の導入や，Few-shot 学習的なアプローチの検討が求められる．

文 献

[1] Chenlu Ding, Jiancan Wu, Yancheng Yuan, Jinda Lu, Kai Zhang, Alex Su, Xiang Wang, and Xiangnan He. 2024. Unified Parameter-Efficient Unlearning for LLMs. CoRR abs/2412.00383 (2024).

[2] Feihu Jin, Jinliang Lu, Jiajun Zhang, and Chengqing Zong. 2023. Instance-Aware Prompt Learning for Language Understanding and Generation. ACM Trans. Asian Low Resour. Lang. Inf. Process (2023).

[3] Wenyu Mao, Jiancan Wu, Weijian Chen, Chongming Gao, Xiang Wang, Xiangnan He. Reinforced Prompt Personalization for Recommendation with Large Language Models. ACM TOIS (2025).

[4] Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. EMNLP, 4222–4235 (2020).

[5] Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. RLPrompt: Optimizing Discrete Text Prompts with Reinforcement Learning. EMNLP, 3369–3391 (2022).

[6] Wang-Cheng Kang and Julian J. McAuley. Self-Attentive Sequential Recommendation. ICDM, 197–206 (2018).

[7] Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. Nature 623, 493–498 (2023).

[8] Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-LLM: A Trainable Agent for Role-Playing. EMNLP, 13153–13187 (2023).

[9] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian J. McAuley, and Wayne Xin Zhao. Large Language Models are Zero-Shot Rankers for Recommender Systems. ECIR, 364–381 (2024).

[10] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS (2022).

[11] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang. Self-Refine: Iterative Refinement with Self-Feedback. NeurIPS (2023).

[12] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate Before Use: Improving Few-shot Performance of Language Models. ICML, 12697–12706 (2021).

[13] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. NeurIPS, 6379–6390 (2017).

[14] Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Yun-Nung Chen, and Kam-Fai Wong. Adversarial advantage actor-critic model for task-completion dialogue policy learning. ICASSP, 6149–6153 (2018).

[15] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, Depeng Jin. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. WWW, 2980–2991 (2021).

[16] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. ICML, 41–48 (2009).

[17] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. UAI, 452–461 (2009).
[18] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. ACM TiiS 5, 4 (2016).
[19] Jianmo Ni, Jiacheng Li, and Julian J. McAuley. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. EMNLP, 188–197 (2019).
[20] Yelp dataset URL: https://business.yelp.com/data/resources/open-dataset/
[21] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. ACM TOIS, vol.20, no.4, 422–446 (2002).

観測可能な閲覧深度を用いたカルーセルUIのためのランキングバンディット

安田　琢真†　　中村　篤祥††

† 北海道大学工学部情報エレクトロニクス学科
†† 北海道大学大学院情報科学研究院
E-mail: †yastar.tkm83@gmail.com, ††atsu@ist.hokudai.ac.jp

あらまし　Web 推薦システムにおいて，カルーセル UI を用いれば，ユーザの閲覧範囲（閲覧深度）はシステムにより直接観測可能である．よって，閲覧深度を直接観測できない設定である従来のランキングバンディット手法（カスケードモデル，位置ベースモデル等）より効率的な学習が可能と考えられる．本研究では，閲覧深度分布の下で，アイテムの期待クリック数を最大化することを目標とする，閲覧深度観測可能な設定のランキングバンディット問題を定式化し，その問題に対する解法アルゴリズムを提案する．また，評価実験を通して，提案手法の有効性を示す．

キーワード　バンディットアルゴリズム, ランキング学習, 推薦システム, 学習理論

1　はじめに

Web 推薦システムでは，ユーザのフィードバック（クリックや滞在など）を用いて提示順序を逐次最適化するオンライン学習が重要である．検索結果，ニュース，EC サイトの推薦など，ユーザに複数アイテムを順位付きで提示する場面は多く，このような設定はランキングバンディットとして定式化されてきた．ランキングバンディットは，探索（推定評価値の信頼度が低いアイテムの提示）と活用（過去の評価値から高い評価が期待できるアイテムの提示）のバランスをとりながら，限られた観測から提示順序を改善する枠組みとして有用であり，推薦・検索の中核要素の一つである．

従来のランキングバンディット研究では，ユーザが上位から順に閲覧する行動をモデル化したカスケードモデルや，位置による閲覧確率を考慮した位置ベースモデル（Position-Based Model; PBM）など，クリックモデルに基づく定式化が広く用いられてきた [9,10]．これらのモデルでは，一般に位置依存の閲覧確率（examination）が潜在変数として扱われ，観測されるのはクリック（あるいは最初のクリック位置）に限られる．その結果，クリックが発生しなかった位置について，(i) ユーザは閲覧したがクリックしなかった負例なのか，(ii) そもそも閲覧されていない未観測なのかが区別しにくい．さらに，PBM のように位置バイアスとアイテム魅力度を分離して推定する必要がある設定では，推定対象が増える分だけ学習効率が低下し得る．このように，従来の枠組みは実用上の重要な状況をカバーする一方で，閲覧状況が追加で観測できる実システムに対して必ずしも最も情報効率のよい学習を与えるとは限らない．

一方，近年の Web サービスではカルーセル UI（横スワイプ型のリスト）やスクロール可能なリスト UI が広く利用されている．図 1 にカルーセル UI の概念図を示す．これらの UI では，ユーザが実際にどこまで表示領域を閲覧したか（最大表示位置，閲覧深度）が，クライアント側のイベントログ等から直接観測可能な場合がある．閲覧深度が観測できれば，学習に利用すべきサンプルを実際に閲覧された位置に限定できる．すなわち，閲覧深度以下の位置ではクリックの有無が観測されるため，クリックしなかったという負例を確定でき，それより深い位置は未閲覧として学習に混ぜない，という分離が可能になる．このとき，潜在閲覧確率（examination）による追加の不確実性を避けられるため，同じ試行回数でもより効率的にランキングを学習できることが期待される．しかし，閲覧深度が観測可能であることを前提としたランキングバンディットの定式化と理論的整理は十分に進んでいない．

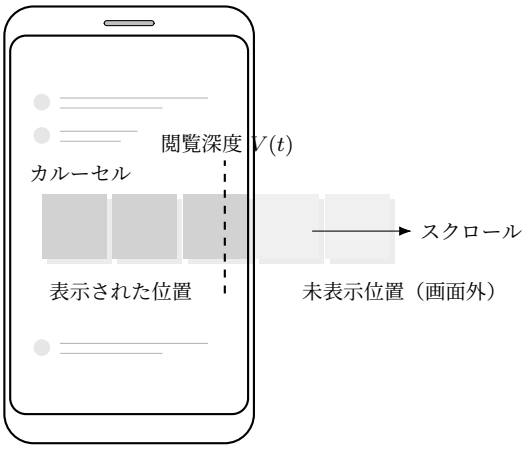

図 1　カルーセル UI の概念図（閲覧深度 $V(t)$ までが観測される）

本研究では，カルーセル UI を想定し，閲覧深度が観測可能な条件下で総クリック数の期待値を最大化するランキングバンディット問題を定式化する．各ラウンドで長さ L のランキングを提示し，ユーザが閲覧した最大位置（閲覧深度）までのクリックのみが観測されるとする．このとき期待報酬は，閲覧深度分布に由来する位置ごとの閲覧確率と，アイテム固有の魅力度（閲覧された場合のクリック率）の積の和として表される．本稿ではまず，閲覧深度観測の効果を明確化するため，クリッ

ク生成をシンプルな形（観測された閲覧深度により閲覧範囲が決まり，閲覧された位置ではアイテムの魅力度に従ってクリックが生起する）で扱う．より複雑な位置効果を同時に含むモデル化は今後の拡張として議論する．

上記の問題に対し，本研究は閲覧深度の観測を直接用いるランキング学習アルゴリズムを提案する．提案法は，そのラウンドで実際に閲覧された位置に置かれたアイテムのみを，露出（観測）されたとみなし，その露出回数に基づいて推定量を更新する．具体的には，露出に基づく信頼区間を用いて探索を行う Observable-Depth UCB と，露出に基づく事後分布からサンプリングする Observable-Depth TS を提示する．さらに，提案 UCB 法についてギャップ依存の期待リグレット上界を導出し，閲覧深度が観測可能であることが学習効率に与える影響を理論的に示す．加えて，シミュレーション実験により，PBM/Cascade 系の既存ベースライン（UCB/TS）と比較して提案法が累積リグレットを改善することを確認する．

本研究の主な貢献は以下の通りである．

- カルーセル UI を想定し，閲覧深度が観測可能なランキングバンディット問題を定式化した．
- 閲覧深度観測に基づき，未観測位置を学習に混ぜないオンライン学習アルゴリズム（Observable-Depth UCB/TS）を提案した．
- 提案 UCB 法についてギャップ依存の期待リグレット上界を与え，深度観測の効果を理論的に整理した．
- シミュレーション実験により，PBM/Cascade 系ベースラインに対する学習効率の改善を示した．

2 関 連 研 究

本節では, 本研究と関係の深い研究領域として, (i) バンディットアルゴリズムの基礎，(ii) クリックモデルに基づくランキングバンディット（カスケードモデル，PBM）と理論保証，(iii) カルーセル等の複数リスト UI（スクロール・閲覧深度）に関する行動モデル化，を概観し，本研究の位置づけを明確化する．

2.1 バンディットアルゴリズムの基礎

バンディットアルゴリズムは，不確実な報酬を持つ選択肢（腕）から逐次的に選択し，累積報酬を最大化するオンライン学習の枠組みである [3,11]．代表的手法として，上側信頼区間（Upper Confidence Bound: UCB）に基づく手法 [1] や事後分布からサンプルして探索する Thompson Sampling [14] などがある．本研究はこれらの枠組みをランキング提示（複数アイテム同時提示）に拡張した設定を扱う．

2.2 クリックモデルとランキング学習

検索・推薦におけるクリックログは，アイテムの真の魅力度だけでなく表示位置や UI の影響を強く受けるため，クリック確率を生成モデルとして記述するクリックモデルが研究されてきた．位置バイアス（position bias）の存在とそのモデル化は古くから議論されており [6]，クリックモデルの体系的な整理として [4] がある．これらのモデルに基づきオンラインでランキングを学習する研究として，以下の 2.3-2.5 節で説明するようなランキングバンディットが発展している．

2.3 カスケードモデルに基づくランキングバンディット

カスケードモデルは，ユーザが上位から順に閲覧し，最初に魅力的なアイテムをクリックするとそこで閲覧を停止する，という逐次閲覧を仮定するクリックモデルである．Kveton らはカスケードモデル下でのランキングバンディットを定式化し，部分観測（最初のクリックまで）に基づく学習アルゴリズムとリグレット解析を与えた [9]．本研究の設定はランキング提示という点では同様であるが，カルーセル UI 等ではどこまで閲覧されたか（閲覧深度）がログから観測可能である点が異なる．この追加情報により，未閲覧位置を負例として誤って扱うことを避けつつ，実際に露出した位置に限定した統計量で推定を進められることが期待される．

2.4 位置ベースモデル（PBM）に基づく複数選択バンディット

位置ベースモデル（Position-Based Model: PBM）は，位置 j の閲覧確率（examination）とアイテムの魅力度の積でクリック確率を表す代表的モデルである．Lagrée らは PBM 下での複数スロット提示（multiple-play）を扱うランキングバンディットを研究し [10]，Komiyama らは更に位置バイアスが未知である設定での理論解析とアルゴリズムを与えた [8]．これらの研究では一般に examination は観測できず，クリックのみから位置バイアスと魅力度を分離推定する必要があるため，推定対象が増える分だけ学習が難しくなり得る．一方，本研究はカルーセル UI 等において閲覧深度が観測可能である状況に着目し，examination を潜在として推定するのではなく，露出（観測）された位置を閲覧深度から直接判定して学習に反映する点で立場が異なる．

従来モデルと本研究の違いは表 1 に整理する．

2.5 カルーセル UI と複数リスト提示

近年の推薦システムでは，単一のランキングリストだけでなく，複数のリスト（カルーセル）を並べる UI が用いられる．このような UI では，ユーザ行動が 2 次元的になり，従来の単一リスト向けクリックモデルでは捉えにくい側面が生じる．Rahdari らはランキングリストからカルーセルへの拡張としてカルーセル向けのクリックモデルを提案し [13]，さらにカルーセル UI を想定したシミュレーション評価の枠組みを検討している [12]．カルーセルとバンディットを組み合わせた応用研究としては，Bendada らが音楽ストリーミングのカルーセル最適化を文脈バンディットとして扱い，実データで有効性を示した [2]．これらは主に行動モデル化・評価手法の側面を中心に扱うのに対し，本研究は閲覧深度が観測可能という実システムで成り立ちやすい情報を前提として，オンラインランキング学習（バンディット）を定式化し，アルゴリズムおよび理論保証（リグレット解析）を与える点に特徴がある．

2.6 本研究の位置づけ

以上を踏まえると，本研究はランキングバンディット（カス

観点	カスケードモデル	位置ベースモデル（PBM）	提案モデル（深度観測）
閲覧深度の観測	観測不可（潜在）	観測不可（潜在）	観測可能（$V(t)$）
クリック観測	最初のクリックまで	位置ごとのクリック	表示位置のクリック
未閲覧位置の扱い	未観測と負例が混在	未観測と負例が混在	未表示として分離
推定対象	魅力度のみ	魅力度と位置バイアス	魅力度のみ（深度は観測）
学習の性質	部分観測で学習	位置バイアスによる補正が必要	露出に基づく更新
代表文献	[9]	[10]	本研究

表 1　既存モデルと提案モデルの比較

ケード／ PBM）という確立した枠組みを踏まえつつ [8–10]，カルーセル UI における閲覧深度の観測可能性に着目して，(1) 露出（観測）に基づく更新則を持つ学習アルゴリズムを設計し，(2) 深度観測を組み込んだ理論解析を与えることで，クリックのみを観測とする従来設定と比べて何が効率化されるかを明確化することを目的とする．従来モデルと本研究の違いは表 1 に整理する．

3　問 題 設 定

3.1　ランキング提示

アイテム集合を $[K] := \{1, 2, \ldots, K\}$，提示枠数（カルーセル内のスロット数）を L とする。各ラウンド $t = 1, 2, \ldots, T$ において，学習者は重複のない長さ L のランキング（順序付きリスト）

$$\boldsymbol{a}(t) = (a_1(t), a_2(t), \ldots, a_L(t)) \in \mathcal{A}$$

を提示する。ここで選択可能なランキングの集合は

$$\mathcal{A} = \left\{(a_1, \ldots, a_L) \in [K]^L \mid a_{j_1} \neq a_{j_2}\ (j_1 \neq j_2)\right\}$$

である。

3.2　閲 覧 深 度

ユーザはラウンド t においてカルーセルをスクロールし，最大で位置 $V(t) \in \{1, \ldots, L\}$ まで閲覧する。本研究では，この閲覧深度 $V(t)$ がクライアントログ等により直接観測可能である設定を扱う。

閲覧深度列 $V(1), V(2), \cdots$ は独立同分布に従うものとし，その分布を

$$p_v := \Pr(V(t) = v), \qquad \sum_{v=1}^{L} p_v = 1$$

とおく。また，位置 j が表示（閲覧範囲に含まれる）される累積確率を

$$P_j := \Pr(V(t) \geqq j) = \sum_{v=j}^{L} p_v$$

と定義すると，以下が成立する。

$$1 = P_1 \geqq P_2 \geqq \cdots \geqq P_L > 0$$

3.3　クリック生成と観測フィードバック

ラウンド t でアイテム i は閲覧されたとき，クリックされたら 1，されなかったら 0 の値をとる確率変数を $C_i(t)$ とする．本稿では確率変数 $C_i(t)$ はベルヌーイ分布に従う，つまり

$$C_i(t) \sim \mathrm{Bernoulli}(\theta_i)$$

とする．ここでアイテム i のクリック率 (本質的魅力度)$\theta_i \in (0, 1)$ は未知パラメータとする．ラウンド t においては，閲覧されたアイテム $a_j(t)$ $(j \leqq V(t))$ に対してのみ，$C_{a_j(t)}(t)$ のフィードバックが観測される．

3.4　報酬と目標

ランキング $\boldsymbol{a}$ を提示したときの総クリック数 $r(\boldsymbol{a})$ をそのラウンドの報酬とする。$r(\boldsymbol{a})$ は

$$r(\boldsymbol{a}) = \sum_{j=1}^{L} \mathbb{1}[j \leqq V(t)] C_{a_j}(t)$$

と書ける。このときの期待報酬は

$$\mathbb{E}[r(\boldsymbol{a})] = \sum_{j=1}^{L} \mathbb{E}\left[\mathbb{1}[j \leqq V(t)] C_{a_j}(t)\right] = \sum_{j=1}^{L} P_j\, \theta_{a_j}$$

となる。

目標は，未知の $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ および与えられた閲覧深度分布の下で，累積期待報酬を最大化するランキング方策を設計することである。

3.5　最適ランキングとリグレット

魅力度 θ_i を降順に並べたときの i 番目の値を $\theta_{(i)}$ とする．$\boldsymbol{a}^\star$ を

$$\boldsymbol{a}^\star = ((1), (2), \ldots, (L))$$

とすると

$$\mathbb{E}[r(\boldsymbol{a}^\star)] = \sum_{j=1}^{L} P_j\, \theta_{(j)} = \max_{\boldsymbol{a} \in \mathcal{A}} \mathbb{E}[r(\boldsymbol{a})]$$

を満たす．つまり $\boldsymbol{a}^\star$ は最適ランキングである．2 番目に最適なランキングを $\boldsymbol{a}^{\star\star}$ で表す．つまり

$$\boldsymbol{a}^{\star\star} = \max_{\boldsymbol{a} \in \mathcal{A}, \mathbb{E}[r(\boldsymbol{a})] < \mathbb{E}[r(\boldsymbol{a}^\star)]} \mathbb{E}[r(\boldsymbol{a})]$$

とする．学習者がラウンド t で選択したランキングを $\boldsymbol{a}(t)$ とすると，期待リグレットは

$$\mathrm{Reg}(T) := \mathbb{E}\left[\sum_{t=1}^{T} \Big(r(\boldsymbol{a}^\star) - r(\boldsymbol{a}(t)) \Big)\right]$$

で定義される。

最適ランキング $\boldsymbol{a}^\star$ に対するランキング $\boldsymbol{a}$ のギャップ $\Delta_{\boldsymbol{a}}$ を以下の通り定義する．

$$\Delta_{\boldsymbol{a}} = \mathbb{E}[r(\boldsymbol{a}^\star)] - \mathbb{E}[r(\boldsymbol{a})] = \sum_{j=1}^{L} P_j(\theta_{(j)} - \theta_{a_j})$$

任意のランキング $\boldsymbol{a}$ に対して $0 \leqq r(\boldsymbol{a}) \leqq L$ であるから，$0 \leqq \Delta_{\boldsymbol{a}} \leqq L$ が成り立つ．このギャップを用いて $\mathrm{Reg}(T)$ は以下のように表現できる．

$$\mathrm{Reg}(T) = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{a}(t)}\right]$$

3.6 既存モデルとの関係

本設定は PBM における位置依存の閲覧確率（examination）を，潜在変数としてではなく観測変数 $V(t)$ として扱う点が特徴である。その結果，表示された位置に置かれた各アイテムについては Bernoulli(θ_i) の独立サンプルが直接得られ，推定と解析が単純化される。

4 提 案 手 法

本節では，第 3 節で定式化した閲覧深度 $V(t)$ が観測可能なランキングバンディットに対し，閲覧深度の観測を直接利用する学習アルゴリズムを提案する．提案法の基本方針は，そのラウンドで実際に表示された（$j \leqq V(t)$）位置に置かれたアイテムのみを露出（観測）として数え，その露出回数に基づき推定を更新することである．これにより，未表示位置（$j > V(t)$）を負例として誤って扱うことを避けつつ，表示された位置に関しては Bernoulli(θ_i) の独立サンプルとして扱って推定することができる．

4.1 統計量（露出回数とクリック回数）

ラウンド t までに，アイテム i が表示（露出）された回数とクリックされた回数を

$$n_i(t) := \sum_{s=1}^{t}\sum_{j=1}^{L} \mathbb{1}\big[a_j(s) = i,\ j \leqq V(s)\big], \tag{1}$$

$$s_i(t) := \sum_{s=1}^{t}\sum_{j=1}^{L} \mathbb{1}\big[a_j(s) = i,\ j \leqq V(s)\big] C_i(s) \tag{2}$$

と定義する．このとき，表示（露出）された位置に限ればクリックは Bernoulli(θ_i) に従うため，$n_i(t) \geqq 1$ のとき，

$$\hat{\theta}_i(t) := \frac{s_i(t)}{n_i(t)}$$

をアイテム魅力度 θ_i の推定量として用いる．

4.2 Observable-Depth UCB（OD-UCB）

UCB 型の探索を行うため，各ラウンド $t \geqq 1$ において

$$\mathrm{UCB}_i(t) := \begin{cases} \hat{\theta}_i(t-1) + \sqrt{\dfrac{\alpha \log t}{n_i(t-1)}} & (n_i(t-1) \geqq 1) \\ +\infty & (n_i(t-1) = 0) \end{cases}$$

Algorithm 1 Observable-Depth UCB

Require: アイテム数 K，スロット数 L，パラメータ $\alpha > 0$

$n_i \leftarrow 0,\ s_i \leftarrow 0 \quad (i = 1, \ldots, K)$
for $t = 1, 2, \ldots, T$ **do**
  for $i = 1, \ldots, K$ **do**
    $\mathrm{UCB}_i \leftarrow \begin{cases} s_i/n_i + \sqrt{\alpha \log t / n_i} & (n_i \geqq 1) \\ +\infty & (n_i = 0) \end{cases}$
  end for
  $\boldsymbol{a}(t) \leftarrow$ UCB スコア上位 L 個を降順に並べたランキング
  ランキング $\boldsymbol{a}(t)$ を提示し，閲覧深度 $V(t)$ とクリック $(C_{a_1(t)}(t), \ldots, C_{a_{V(t)}(t)}(t))$ を観測
  for $j = 1, \ldots, V(t)$ **do**
    $i \leftarrow a_j(t)$
    $n_i \leftarrow n_i + 1$
    $s_i \leftarrow s_i + C_i(t)$
  end for
end for

Algorithm 2 Observable-Depth TS

Require: アイテム数 K，スロット数 L，事前パラメータ $a_0, b_0 > 0$

$n_i \leftarrow 0,\ s_i \leftarrow 0 \quad (i = 1, \ldots, K)$
for $t = 1, 2, \ldots, T$ **do**
  for $i = 1, \ldots, K$ **do**
    $\tilde{\theta}_i \sim \mathrm{Beta}(a_0 + s_i,\ b_0 + n_i - s_i)$
  end for
  $\boldsymbol{a}(t) \leftarrow \tilde{\theta}$ の上位 L 個を降順に並べたランキング
  ランキング $\boldsymbol{a}(t)$ を提示し，閲覧深度 $V(t)$ とクリック $(C_{a_1(t)}(t), \ldots, C_{a_{V(t)}(t)})$ を観測
  for $j = 1, \ldots, V(t)$ **do**
    $i \leftarrow a_j(t)$
    $n_i \leftarrow n_i + 1$
    $s_i \leftarrow s_i + C_i(t)$
  end for
end for

をアイテム $i \in [K]$ のスコアとして用いる．ただし，$\alpha > 0$ は探索と活用のバランスをとるパラメータとする．

そして $\mathrm{UCB}_i(t)$ の大きい順に上位 L 個のアイテムを選び，それらをスコア降順に並べたランキング $\boldsymbol{a}(t)$ を提示する．観測後，深度 $V(t)$ までの位置に置かれたアイテムのみ統計量 (n_i, s_i) を更新する．疑似コードを Algorithm 1 に示す．

4.3 Observable-Depth TS（OD-TS）

次に，Thompson Sampling（TS）に基づくアルゴリズムを与える．Algorithm 2 に疑似コードを示す．

各アイテム i に対し，事前分布として $\theta_i \sim \mathrm{Beta}(a_0, b_0)$ を仮定し，露出回数 $n_i(t)$ とクリック回数 $s_i(t)$ に基づいて事後分布

$$\theta_i \mid \mathcal{H}_t \sim \mathrm{Beta}\big(a_0 + s_i(t),\ b_0 + n_i(t) - s_i(t)\big)$$

を得る．各ラウンド t で各アイテムから独立にサンプル $\tilde{\theta}_i(t) \sim \mathrm{Beta}(a_0 + s_i(t-1),\ b_0 + n_i(t-1) - s_i(t-1))$ を生成し，$\tilde{\theta}_i(t)$ の大きい順に上位 L 個を選んで提示する．更新

は UCB と同様に $j \leqq V(t)$ の位置に限って行う．

4.4 計 算 量

各ラウンドで全アイテムのスコアを計算し，上位 L 個を選ぶため，ヒープを用い入れば計算量は $O(K \log L)$ である．また，統計量の更新の計算量は $O(V(t))$ であり，$V(t) \leqq K$ であるから，OD-UCB，OD-TS 共に [1]，1 ラウンドの計算量は $O(K \log L)$ である．

5 リグレット解析

本節では，提案手法 **Observable-Depth UCB**（Algorithm 1）の期待リグレット上界を与える．解析の骨格は古典的な UCB 解析（集中不等式＋劣腕が選ばれる回数の上界）に従う [1, 11]．一方で本設定では，閲覧深度 $V(t)$ が観測されるため，(i) $j \leqq V(t)$ の位置に置かれたアイテムのみが露出（観測）される，(ii) 表示回数（ランキングに含めた回数）と露出回数が一致しないという点が通常の PBM/Cascade の解析と異なる [8–10]．

最適集合（魅力度上位 L 個）を

$$\mathrm{Top}_L := \{(i) \in [K] \mid i = 1, 2, \ldots, L\}$$

と定義する．

推定値 $\hat{\theta}_i(t) := s_i(t)/n_i(t)$ に対し，良い事象

$$\mathcal{E}_t := \bigcap_{i=1}^{K} \left\{ \left|\hat{\theta}_i(t) - \theta_i\right| \leqq \sqrt{\frac{\alpha \log t}{n_i(t)}} \right\}$$

を導入する．事象 $\mathcal{E}_t$ の余事象を $\mathcal{E}_t^c$ で表す．

定理 1. $\alpha > 1/2$ とする．*OD-UCB* に関し，任意の自然数 $1 \leqq d \leqq L$ に対して

$$\begin{aligned}\mathrm{Reg}(T) \leqq\ & 32\alpha \log T \left(\frac{\left(\sum_{k=1}^{L} P_k\right)^2}{d} + \left(\sum_{k=1}^{d} P_k\right)^2 \right) \\ & \times \left(\sum_{i=L+1}^{K} \frac{1}{P_L(\theta_{(L)} - \theta_{(i)})} + \frac{L}{\Delta_{\boldsymbol{a}^{\star\star}}} \right) \\ & + 2LK\,\zeta(2\alpha).\end{aligned}$$

ここで $\zeta(s) := \sum_{t=1}^{\infty} t^{-s}$ はリーマンゼータ関数であり，$s > 1$ で収束する．特に $\alpha > 1/2$ のとき $2\alpha > 1$ より $\sum_{t=1}^{\infty} t^{-2\alpha} = \zeta(2\alpha)$ が有限である．

Proof. 事象 $\mathcal{E}_t$，$\mathcal{E}_t^c$ を用いて $\mathrm{Reg}(T)$ は以下のように表現できる．

$$\mathrm{Reg}(T) = \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{a}(t)} \mathbb{1}[\mathcal{E}_t]\right] + \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{a}(t)} \mathbb{1}[\mathcal{E}_t^c]\right]$$

第 1 項を補題 5 で，第 2 項を補題 1 で上から抑えることにより，不等式は導かれる． □

1：OD-TS に関しては，Beta 分布からサンプリングするのにかかる計算量は無視するものとする．

補題 1. $\alpha > 1/2$ のとき，以下の不等式が成り立つ．

$$\mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{a}(t)} \mathbb{1}[\mathcal{E}_t^c]\right] \leqq 2LK \sum_{t=1}^{\infty} t^{-2\alpha} = 2LK\,\zeta(2\alpha).$$

Proof. $\Delta_{\boldsymbol{a}} \leqq L$ が任意のランキング $\boldsymbol{a}$ に対して成り立つので

$$\mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{a}(t)} \mathbb{1}[\mathcal{E}_t^c]\right] \leqq L \sum_{t=1}^{T} \Pr\left(\mathcal{E}_t^c\right)$$

が成り立つ．$\Pr\{\mathcal{E}_t^c\}$ は，以下の計算より $2Kt^{-2\alpha}$ で上から抑えられることがわかる．

$$\begin{aligned}\Pr(\mathcal{E}_t^c) =& \Pr\left\{ \bigcup_{i=1}^{K} \left\{ \left|\hat{\theta}_i(t) - \theta_i\right| > \sqrt{\frac{\alpha \log t}{n_i(t)}} \right\} \right\} \\ \leqq& \sum_{i=1}^{K} \Pr\left\{ \left|\hat{\theta}_i(t) - \theta_i\right| > \sqrt{\frac{\alpha \log t}{n_i(t)}} \right\} \\ =& \sum_{i=1}^{K} \Pr\left\{ \left\{ \hat{\theta}_i(t) > \theta_i + \sqrt{\frac{\alpha \log t}{n_i(t)}} \right\} \right. \\ & \qquad \left. \cup \left\{ \hat{\theta}_i(t) < \theta_i - \sqrt{\frac{\alpha \log t}{n_i(t)}} \right\} \right\} \\ \leqq& \sum_{i=1}^{K} \Pr\left\{ \hat{\theta}_i(t) > \theta_i + \sqrt{\frac{\alpha \log t}{n_i(t)}} \right\} \\ & + \sum_{i=1}^{K} \Pr\left\{ \hat{\theta}_i(t) < \theta_i - \sqrt{\frac{\alpha \log t}{n_i(t)}} \right\} \\ \leqq& \sum_{i=1}^{K} e^{-2n_i(t) \cdot \frac{\alpha \log t}{n_i(t)}} + \sum_{i=1}^{K} e^{-2n_i(t) \cdot \frac{\alpha \log t}{n_i(t)}} \\ =& 2Ke^{-2\alpha \log t} = 2Kt^{-2\alpha}\end{aligned}$$

ただし，最後の不等号は Hoeffding 不等式を用いた．よって

$$\sum_{t=1}^{T} \Pr(\mathcal{E}_t^c) \leqq 2K \sum_{t=1}^{\infty} t^{-2\alpha}$$

であり，$\alpha > 1/2$ のとき右辺は収束するので補題が成り立つ． □

事象 $\mathcal{F}_t$ を

$$\mathcal{F}_t = \left\{ 0 < \Delta_{\boldsymbol{a}(t)} \leqq 2 \sum_{j=1}^{L} P_j \sqrt{\frac{\alpha \log t}{n_{a_j(t)}(t-1)}} \right\}$$

と定義すると以下の補題が成り立つ．

補題 2. 以下の包含関係が成り立つ．

$$\mathcal{E}_t \cap \{\Delta_{\boldsymbol{a}(t)} > 0\} \subseteq \mathcal{F}_t$$

Proof. ラウンド t においてランキング $\boldsymbol{a}(t)$ が選ばれているので，

$$\sum_{j=1}^{L} P_j \mathrm{UCB}_{a_j(t)}(t) \geqq \sum_{j=1}^{L} P_j \mathrm{UCB}_{(j)}(t)$$

が成り立っている．このとき事象 $\mathcal{E}_t$ が起こっているとする

と，すべての $j \in [K]$ に対して $\theta_j - \sqrt{\frac{\alpha \log t}{n_j(t-1)}} \leqq \hat{\theta}_j(t-1) \leqq \theta_j + \sqrt{\frac{\alpha \log t}{n_j(t-1)}}$ が成り立っているので，

$$\sum_{j=1}^{L} P_j \left(\theta_{a_j(t)} + 2\sqrt{\frac{\alpha \log t}{n_{a_j(t)}(t-1)}} \right) \geqq \sum_{j=1}^{L} P_j \theta_{(j)}$$

が成り立つ．したがって

$$\Delta_{\boldsymbol{a}(t)} = \sum_{j=1}^{L} P_j \theta_{(j)} - \sum_{j=1}^{L} P_j \theta_{a_j(t)} \leqq 2 \sum_{j=1}^{L} P_j \sqrt{\frac{\alpha \log t}{n_{a_j(t)}(t-1)}}$$

が成り立つ．よって事象 $\mathcal{E}_t$ と事象 $\Delta_{\boldsymbol{a}(t)} > 0$ が同時に起こっていれば事象 $\mathcal{F}_t$ が起こっている． □

ラウンド t の位置集合 S_t を以下のように定義する．

$$S_t = \left\{ j \in [L] \mid n_{a_j(t)}(t) \leqq \frac{16\alpha(\sum_{k=1}^{L} P_k)^2 \log t}{\Delta^2_{\boldsymbol{a}(t)}} \right\}$$

このとき，自然数 $0 < d < L$ に対し，以下の 2 つの事象を考える．

$$\mathcal{G}_t(d) = \{|S_t| \geqq d\}$$

$$\mathcal{H}_t(d) = \left\{ \begin{array}{l} |S_t| < d, \exists j \in [L], \\ \\ n_{a_j(t)}(t) \leqq \dfrac{16\alpha(\sum_{k=1}^{d} P_k)^2 \log t}{\Delta^2_{\boldsymbol{a}(t)}} \end{array} \right\}$$

このとき，以下の補題が成立する．

補題 3. 任意の自然数 $0 < d < L$ に対し，以下の包含関係が成り立つ．

$$\mathcal{F}_t \subseteq \mathcal{G}_t(d) \cup \mathcal{H}_t(d)$$

Proof. 背理法で証明する．$\mathcal{F}_t$ が成り立っているのに $\mathcal{G}_t(d) \cup \mathcal{H}_t(d)$ が成り立っていないとする．$\mathcal{G}_t(d) \cup \mathcal{H}_t(d)$ が成り立っていないので

$$|S_t| < d, \forall j \in [L], n_{a_j(t)}(t) > \frac{16\alpha(\sum_{k=1}^{d} P_k)^2 \log t}{\Delta^2_{\boldsymbol{a}(t)}}$$

が成り立つ．よって

$$\begin{aligned}
\Delta_{\boldsymbol{a}(t)} \leqq & 2 \sum_{j=1}^{L} P_j \sqrt{\frac{\alpha \log t}{n_{a_j(t)}(t-1)}} && (\mathcal{F}_t \text{ 成立より}) \\
= & 2 \sum_{j \in S_t} P_j \sqrt{\frac{\alpha \log t}{n_{a_j(t)}(t-1)}} + 2 \sum_{j \in [L] \setminus S_t} P_j \sqrt{\frac{\alpha \log t}{n_{a_j(t)}(t-1)}} \\
< & 2 \sum_{j \in S_t} P_j \frac{\Delta_{\boldsymbol{a}(t)}}{4 \sum_{k=1}^{d} P_k} + 2 \sum_{j \in [L] \setminus S_t} P_j \frac{\Delta_{\boldsymbol{a}(t)}}{4 \sum_{k=1}^{L} P_k} \\
= & \frac{\Delta_{\boldsymbol{a}(t)}}{2} \left(\frac{\sum_{j \in S_t} P_j}{\sum_{k=1}^{d} P_k} + \frac{\sum_{j \in [L] \setminus S_t} P_j}{\sum_{k=1}^{L} P_k} \right) \\
\leqq & \Delta_{\boldsymbol{a}(t)}
\end{aligned}$$

という矛盾が導かれる．したがって $\mathcal{F}_t \subseteq \mathcal{G}_t(d) \cup \mathcal{H}_t(d)$ が成り立つ． □

アイテム $i \in [K]$ を含むランキングの集合を $\mathcal{A}_i$ で表す．以下の既存の補題を使って，補題 5 を証明する．

補題 4 (Lemma 2 in Appedix B.4 [5])**.** 任意のアイテム $i \in [K]$，正数 $C > 0$ に対し，以下の不等式が成り立つ．

$$\sum_{t=1}^{T} \mathbb{1} \left[i \in \boldsymbol{a}(t), n_i(t) \leqq \frac{C}{\Delta^2_{\boldsymbol{a}(t)}} \right] \Delta_{\boldsymbol{a}(t)} \leqq \frac{2C}{\Delta_{\min,i}}$$

ただし，$\Delta_{\min,i} = \min_{\boldsymbol{a} \in \mathcal{A}_i \setminus \{\boldsymbol{a}^\star\}} \Delta_{\boldsymbol{a}}$ とする．

事象 $\mathcal{G}_t(d)$ と $\mathcal{H}_t(d)$ を以下のようにさらに細かく分ける．

$$\mathcal{G}_{i,t}(d) = \mathcal{G}_t(d) \cap \{i \in S_t\}$$

$$\mathcal{H}_{i,t}(d) = \mathcal{H}_t(d) \cap \left\{ i \in \boldsymbol{a}(t), n_i(t) \leqq \frac{16\alpha(\sum_{k=1}^{d} P_k)^2 \log t}{\Delta^2_{\boldsymbol{a}(t)}} \right\}$$

すると，

$$\sum_{i=1}^{K} \mathbb{1}[\mathcal{G}_{i,t}(d)] = \mathbb{1}[\mathcal{G}_t(d)] \sum_{i=1}^{K} \mathbb{1}[i \in S_t] \geqq d \mathbb{1}[\mathcal{G}_t(d)] \tag{3}$$

$$\mathbb{1}[\mathcal{H}_t(d)] \leqq \sum_{i=1}^{K} \mathbb{1}[\mathcal{H}_{i,t}(d)] \tag{4}$$

が成り立つ．不等式 (3)， (4) と補題 4 を用いて以下の補題 5 が証明できる．

補題 5. 以下の不等式が成り立つ．

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{a}(t)} \mathbb{1}[\mathcal{E}_t] \right] \leqq & 32\alpha \log T \left(\frac{(\sum_{k=1}^{L} P_k)^2}{d} + \left(\sum_{k=1}^{d} P_k \right)^2 \right) \\
& \times \left(\sum_{i=L+1}^{K} \frac{1}{P_L(\theta_{(L)} - \theta_{(i)})} + \frac{L}{\Delta_{\boldsymbol{a}^{\star\star}}} \right)
\end{aligned}$$

Proof. 以下の式変形により補題が証明できる．

$$\begin{aligned}
& \mathbb{E}\left[\sum_{t=1}^{T} \Delta_{\boldsymbol{a}(t)} \mathbb{1}[\mathcal{E}_t] \right] \\
= & \sum_{t=1}^{T} \mathbb{E}\left[\Delta_{\boldsymbol{a}(t)} \mathbb{1}[\mathcal{E}_t, \Delta_{\boldsymbol{a}(t)} > 0] \right] \\
\leqq & \sum_{t=1}^{T} \mathbb{E}\left[\Delta_{\boldsymbol{a}(t)} \mathbb{1}[\mathcal{F}_t] \right] && (\text{補題 2 より}) \\
\leqq & \sum_{t=1}^{T} \mathbb{E}\left[\Delta_{\boldsymbol{a}(t)} \left(\mathbb{1}[\mathcal{G}_t(d)] + \mathbb{1}[\mathcal{H}_t(d)] \right) \right] && (\text{補題 3 より}) \\
\leqq & \sum_{t=1}^{T} \mathbb{E}\left[\Delta_{\boldsymbol{a}(t)} \left(\frac{1}{d} \sum_{i=1}^{K} \mathbb{1}[\mathcal{G}_{i,t}(d)] + \sum_{i=1}^{K} \mathbb{1}[\mathcal{H}_{i,t}(d)] \right) \right] \\
& && ((3), (4) \text{ より}) \\
= & \mathbb{E}\left[\frac{1}{d} \sum_{i=1}^{K} \sum_{t=1}^{T} \mathbb{1}[\mathcal{G}_{i,t}(d)] \Delta_{\boldsymbol{a}(t)} + \sum_{i=1}^{K} \sum_{t=1}^{T} \mathbb{1}[\mathcal{H}_{i,t}(d)] \Delta_{\boldsymbol{a}(t)} \right] \\
\leqq & \mathbb{E}\left[\frac{1}{d} \sum_{i=1}^{K} \sum_{t=1}^{T} \mathbb{1}\left[i \in \boldsymbol{a}(t), n_i(t) \leqq \frac{16\alpha(\sum_{k=1}^{L} P_k)^2 \log T}{\Delta^2_{\boldsymbol{a}(t)}} \right] \Delta_{\boldsymbol{a}(t)} \right]
\end{aligned}$$

$$+\sum_{i=1}^{K}\sum_{t=1}^{T}\mathbb{1}\left[i\in\boldsymbol{a}(t), n_i(t)\leqq\frac{16\alpha(\sum_{k=1}^{d}P_k)^2\log T}{\Delta_{\boldsymbol{a}(t)}^2}\right]\Delta_{\boldsymbol{a}(t)}\Bigg]$$
$$\leqq 32\alpha\log T\left(\frac{(\sum_{k=1}^{L}P_k)^2}{d}+\left(\sum_{k=1}^{d}P_k\right)^2\right)$$
$$\times\left(\sum_{i=L+1}^{K}\frac{1}{P_L(\theta_{(L)}-\theta_{(i)})}+\frac{L}{\Delta_{\boldsymbol{a}^{\star\star}}}\right)\qquad\text{(補題 4 より)}$$

□

注意 1. *Lagrée* ら *[10]* は，位置ベースモデル *(PBM)* における複数スロット提示（*multiple-play*）を扱うランキングバンディットを研究し, 位置 j の閲覧確率（*examination*） P_j が既知の下で *PBM-UCB* の対数リグレット上界を与えた．同論文の *Theorem 9* によれば，任意の $\epsilon>0$ に対し，ある定数 $C_0(\varepsilon)$ が存在し，任意の自然数 $1\leqq d\leqq L$ に対して

$$\mathrm{Reg}(T)\leqq 16(1+\varepsilon)\log T\left(\frac{\left(\sum_{k=1}^{L}P_k\right)^2}{d}+\left(\sum_{k=1}^{d}P_k\right)^2\right)/P_L^2$$
$$\times\left(\sum_{i=L+1}^{K}\frac{1}{P_L(\theta_{(L)}-\theta_{(i)})}+\frac{L}{\Delta_{\boldsymbol{a}^{\star\star}}}\right)$$
$$+C_0(\varepsilon)\qquad(5)$$

が成り立つ．この式において $\epsilon\to 0$ とした式と定理 *1* で $\alpha\to 1/2$ とした式において, 主要項である第 1 項は P_L^{-2} 倍だけ *PBM-UCB* の方が大きく，表示リスト末尾の閲覧確率 P_L が小さい状況では上界が大きく悪化し得る．

一方，本研究の設定では，各ラウンドでどこまで表示されたかを表す閲覧深度 $V(t)$ が観測できるため，位置 j が露出したかどうか（$V(t)\geqq j$）が直接わかる．これは *PBM* における *examination* が 潜在（*censored*） ではなく 観測（*uncensored*）である状況に対応し，*Lagrée* らも *uncensored PBM* では情報量が単純化することを指摘している *[10]*.

6 実　　験

本節では，提案手法（Observable-Depth UCB/TS；OD-UCB/OD-TS）の有効性を (i) 合成シミュレーション，(ii) 実ログ（RecGaze）から推定したパラメータに基づく実データ駆動実験の 2 つで検証する．いずれも真の環境は第 3 節の閲覧深度つき cutoff 型モデルとし，提案法は各ラウンドで閲覧深度 $V(t)$ を観測できる一方，既存ベースラインは $V(t)$ を観測できない（クリックのみ）という条件で比較する．

6.1 評価指標（累積擬似リグレット）

最適ランキング $\boldsymbol{a}^\star$ に対するアルゴリズムのランキング列 $\boldsymbol{a}(1),\dots,\boldsymbol{a}(T)$ の累積期待リグレット $\sum_{t=1}^{T}\Delta_{\boldsymbol{a}(t)}$ を用いる．合成シミュレーションでは真のパラメータ (θ,P) が既知であるため，各ラウンドの期待報酬差 $\Delta_{\boldsymbol{a}(t)}$ を直接計算して累積する．また，実データ駆動実験ではログから計算した (クリック割合, 閲覧割合) を (θ,P) として用い，合成シミュレーションと同様に各ラウンドの期待報酬差を計算して累積する．

6.2 合成シミュレーション環境

各ラウンド t で閲覧深度 $V(t)\in\{1,\dots,L\}$ を, 以下で述べる分布に従ってサンプルし，$j\leqq V(t)$ の位置のみが露出（表示）される．露出された位置 j では，選ばれたアイテム $i=a_j(t)$ を, 以下の方法で生成したパラメータ θ_i のベルヌーイ分布でクリック $C_i(t)$ を生成する．

a） 閲覧深度分布

露出確率 $P_j=\Pr(V(t)\geqq j)$ を与え，$\Pr(V(t)=j)=P_j-P_{j+1}$（$P_{L+1}=0$）で閲覧深度分布を定める．本稿では, 深度が浅い／深い状況を模すため，$L=5$ のとき以下の 2 つの設定を用いる：

$$\text{(shallow)}\ (P_1,\dots,P_5)=(1.0,0.55,0.30,0.15,0.08),$$

$$\text{(deep)}\ (P_1,\dots,P_5)=(1.0,0.80,0.70,0.60,0.50).$$

b） 実 験 設 定

主設定として $K=50$， $L=5$， $T=200{,}000$ とした．アイテムの魅力度 θ_i は，上位 5 個を 0.18, 0.16, 0.14, 0.12, 0.10 とし，残りの 45 個を [0.09,0.02] の両端を含めた等間隔で生成し, 乱数 seed を固定して一様ランダムにシャッフルして各アイテムに割り当てた．さらに，seed を変えた 5 回の独立試行を行い，平均と標準誤差で評価する．UCB 系アルゴリズムの信頼半径の係数は $\alpha=0.5$ とした．

6.3 実データ駆動実験

公開データセット RecGaze のログを用い，(i) 露出確率列 P（閲覧深度分布）と (ii) アイテム魅力度 θ をデータから算出される閲覧割合，クリック割合に設定し，そのモデル上で各手法の期待リグレットを比較する．本実験の狙いは，閲覧深度観測により P_L が小さい（末尾が見られにくい）状況で理論的に有利となるという定理 1 の含意が，実ログ由来の分布でも観察されるかを確認する点にある．

RecGaze は，カルーセル型 UI におけるユーザ行動を対象としたデータセットであり，視線（eye tracking），クリック，カーソル移動，および選択理由の説明を含む包括的なフィードバックが含まれる．3 つの映画選択タスクにおいて，各ユーザに対して 40 種類のカルーセル画面を提示し，合計 87 名・3,477 回のインタラクションが記録されている [7].

a） 閲 覧 割 合

視線, カーソル移動, クリック等のログから計算した閲覧割合は以下であった：

$$(P_1,\dots,P_{15})=(1.0,0.9031,0.8529,0.7720,0.6879,$$
$$0.2992,0.2991,0.2988,0.2986,0.2968,$$
$$0.2576,0.2574,0.2563,0.2531,0.2432).$$

特に $j=5$ から $j=6$ で P_j が大きく低下しており，上位数件は比較的見られるが，それ以降は露出が急減する浅い閲覧が混在していることがわかる．このような状況では，深度を観測

できない手法は未露出の負例混入により推定効率が低下しやすく，深度観測の利点が顕在化すると期待される．

b） 実 験 設 定

K = 150， L = 15， T = 200,000 とした．RecGaze ログから計算した (θ, P) を固定し，各手法は同一のモデルの下で評価した．ここでクリック割合 θ は各アイテムのクリック数と露出数から計算した．さらに，乱数 seed を変えた 5 回の独立試行を行い，平均と標準誤差で評価した．UCB 系アルゴリズムの信頼半径の係数は $\alpha = 0.5$ とした．

6.4 比 較 手 法

合成・実データ駆動の両実験で，環境は常に深度つき cutoff モデルで固定し，アルゴリズム側が $V(t)$ を利用できるかどうかのみを変えて比較する．

a） 提 案 法

提案法は各ラウンドで $V(t)$ を観測し，$j \leqq V(t)$ の位置のみを露出サンプルとして更新する：

- **OD-UCB**：露出回数に基づく推定と UCB により上位 L 件を提示する．
- **OD-TS**：露出回数に基づく Beta 事後分布からサンプルし，上位 L 件を提示する．

b） ベースライン

ベースラインは深度を観測できない状況を想定し，クリック系列のみから学習を行う：

- **PBM-UCB（known bias）**：位置バイアス P_j が既知として PBM の更新を行う UCB．
- **PBM-TS（known bias）**：同様に P_j 既知の PBM 系 TS [10]．事後分布が Beta に閉じず，θ のサンプルは棄却サンプリングで生成するため，OD-TS より計算コストが大きくなりうる．
- **Cascade-UCB（last-click heuristic）**：最後のクリック位置までを露出範囲として更新する単純ベースライン．

6.5 結 果

図 2，図 3 に合成シミュレーション環境（shallow/deep）での累積リグレットを示す．また図 4 に RecGaze 実ログから計算した閲覧割合 P に基づく累積期待リグレットを示す．各曲線は複数 seed の平均であり，誤差帯は標準誤差を表す．

shallow 設定（図 2）では，OD-TS が最小の累積リグレットを達成し，PBM-TS がこれに僅差で続いた．一方で UCB 系は全体に大きく，特に PBM-UCB（閲覧確率 P 既知）は大きなリグレットを示した．OD-UCB は PBM-UCB および Cascade-UCB を大きく上回り，深度観測を用いることが学習効率の改善に寄与していることが確認できる．また Cascade-UCB は試行間のばらつきが相対的に大きい傾向が見られた．

deep 設定（図 3）でも，OD-TS が一貫して最良であり，PBM-TS が次点となった．OD-UCB は TS 系に比べると大きいものの，PBM-UCB および Cascade-UCB より小さい累積リグレットを示し，shallow と同様に深度観測の効果が確認された．

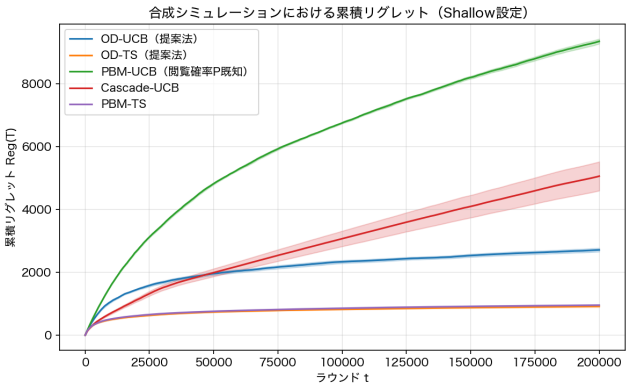

図 2 shallow 設定における累積リグレット．

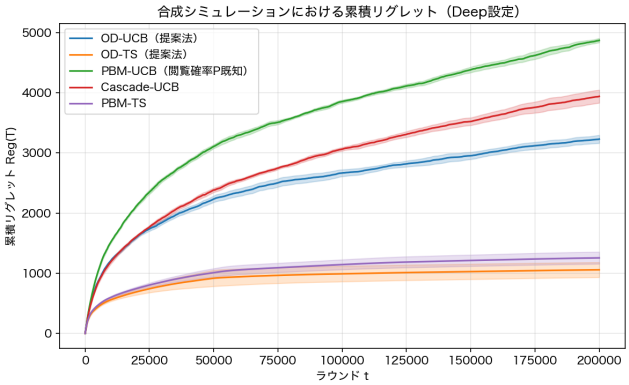

図 3 deep 設定における累積リグレット．

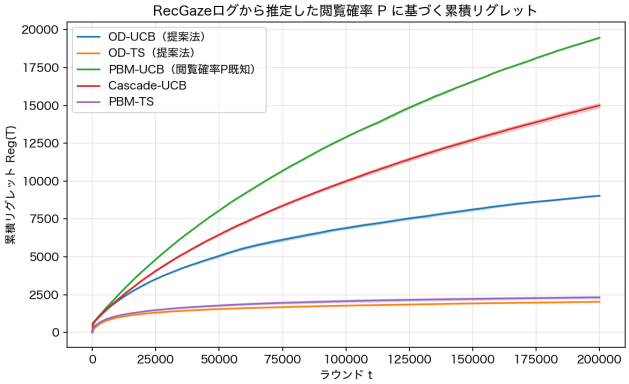

図 4 RecGaze ログから計算した (θ, P) に基づく累積擬似リグレット．

RecGaze を用いた実験（図 4）においても，OD-TS が最小，PBM-TS が次点であり，OD-UCB は PBM-UCB および Cascade-UCB を大きく上回った．特に PBM-UCB は最も大きい累積擬似リグレットを示し，閲覧確率が既知であっても深度を観測しない学習は不利になり得ることが示唆される．

6.6 考 察

提案法（OD-UCB/OD-TS）が改善する主因は，観測された深度 $V(t)$ までの位置のみを学習に用いることで，未露出位置をクリック 0 の負例として混入させない点にある．この効果は shallow 設定および RecGaze を用いた実験で特に顕著であり（図 2, 4），深部位置の露出が希薄な状況では深度を観測できな

い手法ほど負例混入が増え，推定効率が低下しやすい．

一方で deep 設定では多くの位置が露出されるため，深度観測による利得は shallow に比べて相対的に小さくなり得るが，本実験ではそれでも OD-UCB が PBM-UCB や Cascade-UCB を一貫して上回った（図 3）．これは，露出が増えてもどこまで見られたかという情報を明示的に用いることで，学習に用いるサンプルの質（負例混入の抑制）が改善されるためと解釈できる．

また，TS 系（OD-TS / PBM-TS）が UCB 系より大幅に小さい累積リグレットを示した点も重要である．特に OD-TS は shallow/deep/RecGaze の全条件で最良であり，深度観測による情報を活かしつつ事後分布に基づく探索が有効に機能したことを示唆する．一方，OD-UCB は提案枠組みにより PBM-UCB や Cascade-UCB を改善するものの，TS 系ほどの改善幅は得られていないため，有限時間での信頼半径の保守性や探索強度の設計が性能差として現れている可能性がある．

また，第 5 節で述べたように，定理 1 の主要項は露出確率列 P を通じて $\left(\sum_{k=1}^{L} P_k\right)^2$ や $\sum_{k=1}^{d} P_k$，およびギャップ項に現れる $1/P_L$ に依存する．したがって，末尾閲覧確率 P_L が小さいほど（末尾が見られにくいほど），深度観測により未露出位置の負例混入を抑える利得が大きくなり，深度観測なしの既存手法との性能差（特に OD-UCB と PBM-UCB の差）が拡大しやすい．実際，shallow 設定では P_L が相対的に小さいため差が顕著であり（図 2），deep 設定でも shallow ほどではないが末尾確率の有限性に起因する差が残る（図 3）．RecGaze では中位以降で露出確率が大きく低下しており，末尾が見られにくい条件が含まれるため，深度観測の利点が実験でも確認された（図 4）．

RecGaze の結果は推定モデル上のリグレットであり，真のユーザ行動モデルに対する厳密なリグレット保証を与えるものではない．しかし，ログ推定により得られる浅い閲覧を含む条件下で，提案手法（OD-UCB/OD-TS）が一貫して小さい累積期待リグレットを示したことは，深度観測により学習効率が改善するという理論的示唆と整合する．

7 おわりに

本研究では，カルーセル UI を想定し，ユーザの閲覧深度（最大表示位置）が観測可能な条件下でのランキングバンディット問題を定式化した．従来のクリックモデル（カスケードモデル，PBM 等）では閲覧位置（examination）が潜在変数として扱われることが多く，クリックしなかった観測が負例か未観測か判別しにくい．これに対し本研究は，観測された閲覧深度に基づいて実際に露出（観測）された位置のみで学習を更新することで，未観測位置を誤って負例として扱うことを避け，情報効率のよいランキング学習を可能にする枠組みを提示した．

提案手法として，露出回数に基づく信頼区間を用いる Observable-Depth UCB と，同様に露出回数に基づく事後分布からサンプリングする Observable-Depth TS を提案した．さらに Observable-Depth UCB についてギャップ依存の期待リグレット上界を導出し，閲覧深度が観測可能であることが学習効率に寄与することを理論的に整理した．また，シミュレーション実験により，PBM/Cascade 系の既存ベースライン（UCB/TS）と比較して，提案法が累積リグレットを改善することを確認した．

今後の課題としては，まず理論解析の改善が挙げられる．本稿の上界は最悪の閲覧確率として P_L を用いたため保守的であり，位置ごとの閲覧確率 P_j や実際の配置分布を利用したよりタイトな上界への改良が考えられる．また，Observable-Depth TS の理論保証や，下界（深度観測あり設定のミニマックス下界／ギャップ下界）の導出も重要である．さらに，クリック生成に追加の位置効果や文脈情報を含めた拡張，2 次元カルーセル（行×列）や多様性制約を伴う UI 設計に適合したモデル化など，実システムに近い設定への一般化も今後の発展方向である．

謝　辞

研究室内での議論やコメントを通じて貴重な示唆を頂いたアルゴリズム研究室の皆様に感謝いたします．

文　献

[1] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, Vol. 47, No. 2–3, pp. 235–256, 2002.

[2] Walid Bendada, Guillaume Salha, and Théo Bontempelli. Carousel personalization in music streaming apps with contextual bandits. In *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys '20)*, pp. 420–425, 2020.

[3] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[4] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. *Click Models for Web Search*. Morgan & Claypool Publishers, 2015.

[5] Richard Combes, Mohammad Sadegh Talebi, Alexandre Proutière, and Marc Lelarge. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems 28 (NeurIPS 2015), Montreal, Quebec, Canada, December 7–12, 2015*, pp. 2116–2124, 2015.

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

[7] Santiago de Leon-Martinez, Jingwei Kang, Robert Moro, Maarten de Rijke, Branislav Kveton, Harrie Oosterhuis, and Maria Bielikova. RecGaze: The first eye tracking and user interaction dataset for carousel interfaces. In *SIGIR 2025: 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3702–3711. ACM, July 2025.

[8] Junpei Komiyama, Junya Honda, and Akiko Takeda. Position-based multiple-play bandit problem with unknown position bias. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[9] Branislav Kveton, Csaba Szepesvári, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

[10] Paul Lagrée, Claire Vernade, and Olivier Cappé. Multiple-play bandits in the position-based model. In *Advances in*

Neural Information Processing Systems (NeurIPS), 2016.

[11] Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

[12] Behnam Rahdari, Peter Brusilovsky, and Branislav Kveton. Towards simulation-based evaluation of recommender systems with carousel interfaces. *ACM Transactions on Recommender Systems*, Vol. 2, No. 1, pp. 9:1–9:25, 2024.

[13] Behnam Rahdari, Branislav Kveton, and Peter Brusilovsky. From ranked lists to carousels: A carousel click model. *CoRR*, Vol. abs/2209.13426, , 2022.

[14] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, Vol. 25, No. 3–4, pp. 285–294, 1933.

Robust Recommendation against Shilling Attacks via List Consistency and Counterfactual Neighbor Analysis

Fan Mo †§ Chongxian Chen ‡ Xin Fan ‡ and Hayato Yamana †

† Faculty of Science and Engineering, Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan
‡ Dept. of CSCE, Waseda University 3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555 Japan
§Zozo research, Chiba, Japan
E-mail: † bakubonn@toki.waseda.jp, yamana@yama.info.waseda.ac.jp
‡ fan_xin@fuji.waseda.jp, chenc@toki.waseda.jp

Abstract Shilling attacks pose a critical threat to recommendation systems, where malicious users inject crafted interactions or comments to promote target items. Existing defenses can be categorized into explicit detection of fake users or interactions, and anti-shilling recommendation methods. Recent studies increasingly focus on anti-shilling recommendation, which aims to preserve the consistency of recommendation lists after and before attacks, rather than explicitly detecting malicious users. However, existing anti-shilling methods assume the presence of attacks and focus on malicious suppression, which may lead to additional computational overhead while potentially degrading recommendation performance when no shilling attacks are present in the system. Besides, they lack generality, as the anti-shilling methods are designed for specific recommendation models and training procedures, which limits their applicability and practicality in real-world systems where recommenders are frequently updated and continuously evolving. In this work, we propose a counterfactual-based reranking framework for anti-shilling recommendation. This proposed method improves accuracy by refining noisy rankings via list-based consistency when no shilling attacks are present, and naturally suppresses maliciously injected recommendations when attacks occur. Specifically, we construct a recommendation list similarity graph from top-K outputs of a base recommender, leveraging the collaborative filtering assumption that users with similar preferences receive similar recommended items. This list-level consistency is used to suppress anomalous recommendations weakly supported by similar users. Besides, we introduce a counterfactual neighbor–based analysis that measures the stability of user representations by randomly masking neighbors during training. Users exhibiting large embedding variations are regarded as suspicious, as genuine users typically exhibit stable representations; meanwhile, in the absence of shilling attacks, stable representations contribute to consistent preference learning, therefore improve accuracy.

Keyword graph neural network, collaborative filtering, shilling attack, anti-shilling, recommendation system

1. Introduction

Recommendation systems play a central role in modern online platforms by filtering vast amounts of content and guiding user decision-making. In recent years, a variety of learning-based recommendation have been developed, including neural collaborative filtering (NCF) [1] approaches that model nonlinear user–item interactions, graph-based methods [2] [3] [9] [10] [32] that leverage user–item graphs to propagate collaborative signals and model high-order connectivity.

However, the training of recommendation systems relaying on user-generated interactions makes them particularly vulnerable to shilling attacks, a class of adversarial behaviors in which malicious users inject crafted interactions or reviews to artificially promote target items. Such attacks can significantly distort recommendation results, degrade user experience, and undermine the credibility of the platform.

To solve the problem of shilling attack, existing studies can be broadly categorized into two groups: detection-based methods [4] and anti-shilling-based methods [5] [6]. Detection-based approaches suffer from two inherent limitations. 1) Training an effective detector typically requires a large number of labeled fake users to ensure reliable performance; however, in real-world systems, the proportion of ground-truth fake users is usually extremely low [7], leading to severe class imbalance in the training data; 2) Constructing representative features and collecting sufficient training examples is labor-intensive and may further raise concerns regarding privacy compliance in user profiling [8].

Therefore, in recent years, increasing attention has been directed toward anti-shilling recommendation methods [5] [6]. Instead of explicitly detecting fake users or malicious

interactions, anti-shilling approaches aim to mitigate the impact of shilling attacks on recommendation outcomes, ensuring that the recommendation lists remain stable and reliable even in the presence of malicious behaviors. You et al. [5] pioneered the study of anti-shilling recommendation. They proposed a GCN-based anti-shilling recommendation framework that consists of two stages. The model first predicts the probability of a user being fake, and then integrates the predicted scores into the recommendation process to prevent the propagation of negative impacts caused by shilling attacks. Based on You et al., Mu et al. [6] propose Trust-GRS, which estimates the probabilities of users and items being fake by exploiting training dynamics and interaction frequency anomalies. Specifically, the method identifies suspicious users in early training stages and employs a PageRank-based algorithm, termed Shilling-Rank, to propagate fake probabilities over the user–item graph.

However, their methods suffer from several limitations. 1). These methods lack generality, as existing anti-shilling methods are often tightly coupled with specific recommendation models and training procedures, making them difficult to transfer or deploy across different recommendation models. Specifically, applying these methods requires detailed knowledge of the internal structure of the underlying recommender and corresponding modifications to the recommendation model, which limits their applicability in practical scenarios where the base recommender is fixed, proprietary, or costly to retrain, and therefore cannot be easily extended as a black-box component. 2). Existing anti-shilling methods are designed under the explicit assumption that shilling attacks are present, and thus primarily focus on suppressing malicious behaviors. Therefore, when no shilling attacks are present, existing anti-shilling methods are not explicitly designed to optimize recommendation performance under clean settings, which may introduce extra computational overhead and may potentially affect recommendation results.

In this paper, we propose ConsisRec (consistency-aware recommendation), a post-processing approach that operates on the output of a base recommender system. Given the top-K recommendation outputs of a base recommender, we first construct a candidate list-based similarity graph, inspired by collaborative filtering, where users with similar preferences tend to receive similar recommended items. This list-level consistency provides a robust collaborative signal that enables us to suppress anomalous recommendations that are weakly supported by similar users. Furthermore, we introduce a counterfactual neighbor–based analysis to assess the stability of user representations. By masking neighbors during training, we measure how user embeddings vary under neighborhood perturbations. Users exhibiting large embedding variations are regarded as suspicious, as genuine users typically maintain stable representations. This stability signal is used in a soft and model-independent manner, without explicitly detecting or removing users.

Our framework is scalable and non-intrusive, because it operates purely as a post-processing mechanism on top of any base recommender. When no shilling attacks are present, the proposed reranking mechanism improves recommendation accuracy by refining noisy ranking results through list-based collaborative consistency. When shilling attacks occur, the same mechanism naturally suppresses maliciously injected recommendations, achieving effective anti-shilling robustness without sacrificing performance in benign settings. Our contributions are listed as follows.

- We propose ConsisRec, a scalable and model-agnostic post-processing framework for anti-shilling recommendation, applicable to various recommender models.
- We exploit list-based collaborative consistency via a candidate list similarity graph propagation to refine ranking results.
- We introduce a counterfactual stability signal to softly suppress the influence of suspicious users.
- We will further investigate the dual role of ConsisRec in both clean and adversarial environments. Our preliminary results reveal that even under shilling attacks, ConsisRec achieves higher recommendation accuracy than the base recommender in attack-free settings, indicating that the proposed method not only mitigates the negative impact of malicious interactions but also fundamentally improves representation quality beyond the original model's capability.

The remainder of the paper is organized as follows: Section 2 reviews related work on shilling attacks and anti-shilling recommendation. Section 3 introduces the preliminaries. Section 4 presents the proposed ConsisRec framework. Section 5 describes the experimental setup and reports the experimental results. Finally, Section 6 concludes the paper.

2. Related Work

2.1. Shilling attack

Shilling attacks manipulate recommendation systems by strategically injecting artificial user profiles whose

interaction patterns are deliberately designed to promote target items. Shilling attacks can be categorized into three types according to how the attack profiles are generated: heuristic attacks, neural-network-based attacks, and gradient-based attacks. Heuristic attacks generate fake user profiles by following predefined item selection rules [11] [12]. Popularity attacks construct fake user profiles by interacting with target items and a set of popular items, aiming to maximize overlap with genuine users. Random attacks generate fake profiles by combining interactions on target items with randomly selected filler items, to mimic normal user behavior. Recently, shilling attacks based on neural networks and gradient optimization have gained increasing attention. Neural network–based attacks leverage deep learning models to automatically learn realistic interaction patterns for constructing fake user profiles. For example, PRec [13] formulates shilling attack generation as a reinforcement learning problem, while GOAT [14] adopt generative adversarial networks to synthesize attack profiles. Gradient-based attacks cast shilling attack generation as a bi-level optimization problem, where approximate gradients are exploited to iteratively modify the original data and generate the final attack profiles. Neural network–based and gradient-based attacks make defending against shilling attacks increasingly challenging, as such attacks can adaptively optimize injected interactions to closely mimic genuine user behaviors and exploit model-specific vulnerabilities.

2.2. Shilling Attack Defense

Existing defenses against shilling attacks can be divided into two categories: explicit detection of shilling users or items, and anti-shilling recommendation methods that mitigate the impact of malicious manipulation on recommendation results. Explicit detection has been regarded as one of the most straightforward defenses against shilling attacks [15]. Early studies by Burke et al. trained classifiers using carefully designed features extracted from the rating matrix to identify malicious users [16]. Bhaumik et al., proposed unsupervised detection methods based on clustering and data mining techniques to identify fake profiles by exploiting statistical discrepancies between genuine and malicious data [17]. Wu et al. [18], proposed a probabilistic method to train a Naïve Bayes classifier on labeled data and infer posterior probabilities for unlabeled users. In recent years, graph-based detection methods have attracted increasing attention from the researchers. Li et al.[20] proposed SpDetector, which constructs user and item hypergraphs to extract spectral features capturing high-order interactions, and integrates them with rating prediction errors to accurately distinguish fake users from genuine ones. Zhang et al. [19] proposes a user similarity–based graph convolutional network (USGSAD) for the detection, which jointly model user rating correlation and deviation to identify malicious users without manual feature engineering. Hao et al. [21] modifies the graph structure by reweighting edges. They extracted popularity- and rating-based user features, constructed a weighted user graph, and employed a two-stage scheme with partial labeling and regularized GCN to detect hybrid model-generative shilling attacks. Despite their effectiveness, the above explicit detection-based methods have two limitations: 1) They rely on large amounts of labeled fake users, which are scarce in practice and lead to severe class imbalance. 2) Feature engineering and data collection are labor-intensive and raise privacy concerns.

To solve the problems, You et al. [5] pioneered the concept of anti-shilling recommendation by proposing a GCN-based framework that estimates fake-user probabilities and integrates the predicted scores into the recommendation process to mitigate shilling attacks. Building on this idea, Mu et al. [6] proposed Trust-GRS, which identified suspicious users in early training stages and employs a PageRank-based algorithm, termed Shilling-Rank, to propagate fake probabilities over the user–item graph. However, existing anti-shilling methods are model-specific and assume the presence of shilling attacks, which limits their generality and may introduce unnecessary overhead or performance degradation in clean settings.

3. Preliminary

This section introduces the preliminary knowledge about shilling attacks.

3.1. Recommendation task

$U = \{u_1, u_2, u_3, \dots, u_X\}$ and $I = \{i_1, i_2, i_3, \dots, i_Y\}$ denote the sets of users and items, $N = \{N_{u_x} | 1 \le x \le X\}$ denotes the set of N_{u_x}, and $N_{u_x} = \{i_1, i_2, \dots, i_y\}$ consists of the items checked by user u_x. The goal of the recommendation task is to predict a user's preference over unseen items and recommend those that the user is likely to click in the future. Table 1 provides a summary of the notations used in this paper.

3.2. Attacker's goal

We consider the most common shilling attack setting as You et al.[5] and Mu et al. [6], where the attacker aims to boost the ranking of a set of target items I^T. Specifically, the goal is to make the target items appear in the

Table 1: Notations

Notation	Definition
$e_{u_x}^{k}$	The output embedding of user u_x at k^{th} GCN layer
$e_{i_y}^{k}$	The output embedding of item $\boldsymbol{i_y}$ at k^{th} GCN layer
$\boldsymbol{e_{u_x}}$	The final output embedding produced by the GCN for user $\boldsymbol{u_x}$
$\boldsymbol{e_{i_y}}$	The final output embedding produced by the GCN for item $\boldsymbol{i_y}$
C_{u_x}	candidate item set of user u_x
C_{i_y}	candidate user set of item i_y
γ_{u_x}	risk assessment coefficient of user u_x, ranging from 0 to 1, controls the amount of information aggregated from user u_x during GCN propagation
γ_{i_y}	risk assessment coefficient of item $\boldsymbol{i_y}$, ranging from 0 to 1, controls the amount of information aggregated from item $\boldsymbol{i_y}$ during GCN propagation
r_{u_x}	The risk score of the user u_x, represents the estimated likelihood that the user is fake
r_{i_y}	The risk score of the item i_y, represents the estimated likelihood that the item is a target item

recommendation lists of as many users as possible.

3.3. Attacker's capability

To avoid easy detection, the number of malicious user profiles is limited; by default, same as You et al.[5], the injection rate is set to 1%.

3.4. Defender's knowledge

We assume that the defender only has access to check-in data, without any additional side information. Besides, the defender has no prior knowledge of the specific attack strategies.

4. Proposed method

This section introduces the details of the ConsisRec. As illustrated in Fig. 1, the architecture of the proposed method consists of three parts. Step1: The model initializes user and item embeddings using a Gaussian distribution. The shilling risk r_{u_x}/r_{i_y} of user and item is initialized to zero. Step2: Through risk-aware graph convolution propagation, the model produces the final embedding representations for users and items. At this stage, we first construct a graph structure based on the candidate item sets (Section 4.1). We then perform risk-aware neighbor aggregation, where information from suspicious neighbors is adaptively down-weighted during message passing (Section 4.2). We further apply a counterfactual masking strategy that selectively removes a subset of neighboring nodes to evaluate the stability of user representations (Section 4.3). Users whose embeddings exhibit pronounced sensitivity to such perturbations are regarded as suspicious, whereas genuine users are expected to maintain relatively stable representations under counterfactual graph structures. Step3: Based on the user and item embeddings obtained in the second stage, the model computes preference scores via inner products and generates the final recommendation list.

4.1. Candidate-Induced Graph Construction

For each user u_x, we first produce a candidate item set C_{u_x} using the base model by returning the top-L items, where top-L >>top-K. top-K denotes the number of items in the final recommendation list. This ensures that post-processing module training from a sufficiently rich information to mitigate shilling attacks. Based on these candidate sets, we further construct C_{i_y} for each item i_y, which consists of users whose candidate lists include item i_y.

After that, we construct a user-item graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that connects users and items based on the candidates. Specifically, $\mathcal{V}$ denotes the set of user and item nodes while $\mathcal{E}$ is a edge set, is defined as Eq. 1.

$$\mathcal{E} = \{(u_x, i_y) | u_x \in U, i_y \in C_{u_x}\} \tag{1}$$

4.2. Risk-aware GCN propagation

After constructing the graph structure, at each GCN layer, we design user and item representations by aggregating information from their neighbors, as formulated in Eq. 2.

$$e_{u_x}^{k} = \sum_{i_y \in C_{u_x}} \frac{\gamma_{i_y}}{\sqrt{|C_{u_x}| * |C_{i_y}|}} e_{i_y}^{k-1}$$
$$e_{i_y}^{k} = \sum_{u_x \in C_{i_y}} \frac{\gamma_{u_x}}{\sqrt{|C_{u_x}| * |C_{i_y}|}} e_{u_x}^{k-1} \tag{2}$$

, where $k \in \{1,2,3\}$, $e_{u_x}^{k}$ and $e_{i_y}^{k}$ represent the user and item embeddings produced at the k^{th} GCN layer while $e_{u_x}^{0}$ and $e_{i_y}^{0}$ represent the initial user and item embeddings, respectively. The term $1/\sqrt{|C_{u_x}| * |C_{i_y}|}$ serves as a degree-based normalization factor to stabilize message propagation and mitigate over-smoothing, where $|.|$ denotes the size of the candidate-based neighbor set. γ_{u_x} and γ_{i_y} are risk assessment coefficients ranging from 0 to 1, where 1 indicates a fully trusted neighbor. The corresponding computation is defined as Eq. 3.

$$\gamma_{u_x} = max\,(0, 1 - \sigma(\beta * r_{u_x}))$$
$$\gamma_{i_y} = max\,(0, 1 - \sigma(\beta * r_{i_y})) \tag{3}$$

, where β is a hyperparameter that controls the strength of risk r_{u_x}/r_{i_y} of a user/item node. The risk scores r_{u_x} and r_{i_y} represent the estimated likelihood that a user or an item is involved in shilling attack. σ is sigmoid function, used to smoothly map the estimated risk into the range 0 to 1. As

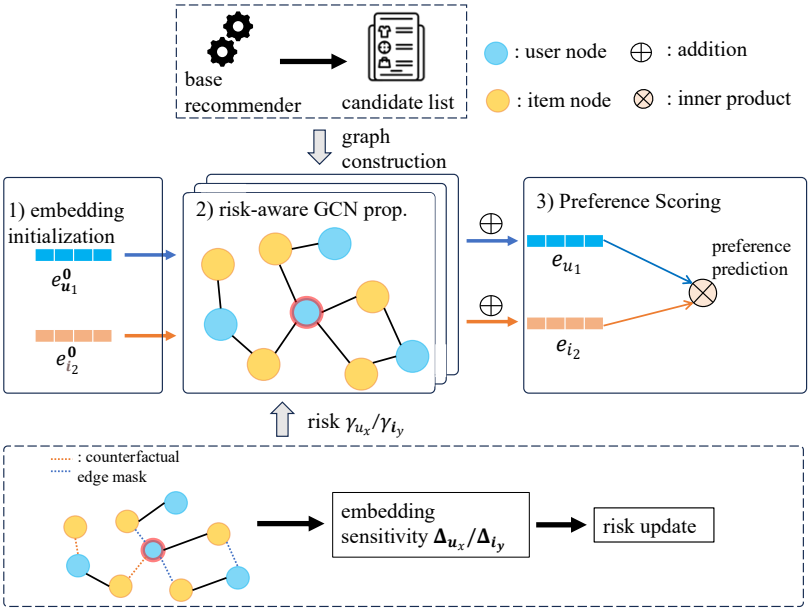

Figure 1: architecture of the proposed method

the estimated risk increases, less information is aggregated from the corresponding node by inversely weighting its contribution during neighbor aggregation. The update details of the risks r_{u_x} and r_{i_y} are described in Section 4.3. After the model outputs the high-order user and item representations, we aggregate all layer's outputs by mean pooling, as Eq. 4.

$$\boldsymbol{e}_{u_x} = \sum_{k=0}^{K} \alpha_k \boldsymbol{e}_{u_x}^{k} \qquad (4)$$
$$\boldsymbol{e}_{i_y} = \sum_{k=0}^{K} \alpha_k \boldsymbol{e}_{i_y}^{k}$$

, where α_k is a weighting coefficient fixed at $1/(K+1)$, with $K=3$. $\boldsymbol{e}_{u_x}$ and $\boldsymbol{e}_{i_y}$ represent the final embedding representations of user $\boldsymbol{u_x}$ and item $\boldsymbol{i_y}$.

4.3. Counterfactual-based risk estimation

This section describes how we quantify and update the risk associated with each user and item based on the counterfactual stability analysis. Specifically, we leverage the sensitivity of user representations to counterfactual neighbor perturbations as a risk signal, where unstable embedding behaviors indicate potential anomalous or unreliable interactions. On the user–item candidate graph $\mathcal{G}$, we first perform S times independent random edge masking operations. For each mask step s, we randomly mask 30% of the edges, treating the corresponding neighbors as absent, and generate the counterfactual graph $\mathcal{G}_s$. On the counterfactual graph $\mathcal{G}_s$, we apply Eqs. 2 to 4 to calculate the user and item embeddings under the counterfactual setting, denoted as $\boldsymbol{e}_{u_x,s}$ and $\boldsymbol{e}_{i_y,s}$.

We then compute the embedding sensitivity magnitude $\boldsymbol{\Delta}_{u_x}/\boldsymbol{\Delta}_{i_y}$ by using l_2 distance as defined in Eq. 5. Alternative distance measures, such as KL divergence, is left for future work.

$$\boldsymbol{\Delta}_{u_x} = \frac{1}{S}\sum_{1}^{S} \left\| \boldsymbol{e}_{u_x} - \boldsymbol{e}_{u_x,s} \right\|_2 \qquad (5)$$
$$\boldsymbol{\Delta}_{i_y} = \frac{1}{S}\sum_{1}^{S} \left\| \boldsymbol{e}_{i_y} - \boldsymbol{e}_{i_y,s} \right\|_2$$

We maintain dynamically updated risk scores r_{u_x} and r_{i_y}. After computing embedding sensitivity magnitude, we update risks using exponential moving average, as Eq. 6.

$$r_{u_x} \leftarrow \boldsymbol{\alpha} r_{u_x} + (1-\boldsymbol{\alpha})\boldsymbol{\Delta}_{u_x} \qquad (6)$$
$$r_{i_y} \leftarrow \boldsymbol{\alpha} r_{i_y} + (1-\boldsymbol{\alpha})\boldsymbol{\Delta}_{i_y}$$

, where $\boldsymbol{\alpha}$ ranges from 0 to 1, controlling the smoothing strength.

4.4. Preference Scoring

Given a user $\boldsymbol{u_x}$ and an item $\boldsymbol{i_y}$ along with their embeddings $\boldsymbol{e}_{u_x}$ and $\boldsymbol{e}_{i_y}$, we compute the preference score using the inner product, which is widely adopted in GCN-based recommendation systems[30] [2], as Eq. 7.

$$\widehat{r_{u_x,i_y}} = \boldsymbol{e}_{u_x}{}^{T} \boldsymbol{e}_{i_y} \qquad (7)$$

We return the top-K items with the highest preference scores from the candidate set as the recommendation list for the user.

4.5. Model training

Pairwise ranking objectives are commonly employed in implicit-feedback recommendation scenarios. Among them, the Bayesian Personalized Ranking (BPR) loss has been extensively used due to its effectiveness in learning personalized ranking signals from implicit interactions [9] [23] [22] [33]. The BPR loss adopts a pairwise optimization strategy by sampling negative items for each observed

user–item interaction and encourages the model to assign higher preference scores to observed (positive) items than to unobserved (negative) ones. Instead of drawing negative items solely from the set of unobserved interactions, we further refine the negative sampling process by excluding items that appear in the candidate set C_{u_x}, same as Mo et al. [10]. The resulting training objective is defined as Eq. 8.

$$L_{BPR} = -\sum_{(u_x, i_y, i_{y'})\epsilon D} ln\sigma\left(\widehat{r_{u_x,i_y}} - \widehat{r_{u_x,i_{y'}}}\right) + \mu\|\boldsymbol{\omega}\|^2 \quad (8)$$

, where $D = \{(u_x, i_y, i_{y'})|i_y \epsilon N_{u_x}, i_{y'} \epsilon N - N_{u_x} - C_{u_x}\}$. The parameter set $\boldsymbol{\omega}$ includes all learnable variables of the model, while μ controls the magnitude of the l_2 regularization term used to alleviate overfitting. Other loss terms, such as regularizing the embedding sensitivity magnitude $\boldsymbol{\Delta}_{u_x}/\boldsymbol{\Delta}_{i_y}$, are left for future work.

5. Preliminary Experiment

5.1. Dataset

We conduct experiments on the ML-1M dataset[24] [31], which contains 6,040 users, 3,952 items and 1,000,209 ratings. Data preprocessing is performed by ARLib [24]. The dataset is randomly split into 70% for training, 10% for tuning, and 20% for testing. We directly use the preprocessed datasets provided by ARLib. The dataset statistics are summarized in Table 2.

5.2. Base Recommender System

We choose LightGCN[2], a representative and widely adopted recommender, as the backbone model.

5.3. Attack and Defense Method

With the help of ARLib, we conducted preliminary experiments to validate the effectiveness of the proposed method under random attacks. We used the recommendation results of the base LightGCN model as the baseline for comparison.

5.4. Hyperparameter setting

Our method involves several important hyperparameters, including the size of the candidate set generated by the base recommender (top-L), the exponent β that controls the strength of risk-aware aggregation, the number of masking operations S, and the risk update exponent α. We set the candidate list size top-L to be twice the final recommendation length, i.e., top-L =40. β is set to 0.7. The number of counterfactual masking operations S is set to 3. The risk update smoothing exponent α is set to 0.9.

5.5. Metric

Following You et al [5], we recommend 20 items for each user. To evaluate recommendation accuracy, we adopt commonly used metrics, including Recall@20, and NDCG@20. Comparing recommendation accuracy with the base recommender validates the effectiveness of our method. Besides, to directly evaluate anti-shilling effectiveness, we measure the Target Item Exposure (TIE), defined as the number of times the target item appears in users' recommendation lists, where a lower value indicates better defense performance.

In this preliminary study, we compare our method with the base recommender in terms of recommendation accuracy. The evaluation with additional metrics TIE, more datasets, additional recommender models (e.g., NCF [1], NFCG[25], NCL [26], SimGCL[27]), stronger attack strategies (e.g., DLAttack[28], GOAT[14], Pipattack[29]), comparisons with state-of-the-art anti-shilling defenses(e.g., Anti-fakeu [5], Trust-GRS[6]), and directly applying ConsisRec to the base recommender model are left for future work. Before launching the attack, we randomly select five unpopular items as target items and generate fake users accounting for 1% of the number of normal users.

5.6. Experimental Results

In this section, we conduct a comparative evaluation between the proposed method and baseline methods. Table 3 presents the experimental results.

5.7. Comparison between ConsisRec with baselines

Compare LightGCN+RandomAttack with LightGCN, under random attack settings, the performance of the base recommender is affected by maliciously injected interactions. By contrast, even in the presence of shilling attacks, ConsisRec achieves notable improvements over LightGCN, yielding relative gains of 2.11% in Recall and 2.35% in NDCG. These results demonstrate that ConsisRec can naturally suppress anomalous recommendations introduced by shilling attacks, while preserving and even enhancing recommendation accuracy. Notably, ConsisRec treats the base recommender as a black box and relies solely on the candidate sets it generates, without modifying the internal structure or parameters of the base model, which ensures the proposed framework with high generality and makes it readily applicable to various recommender systems.

Table 2: Dataset Statistics

Dataset	#user	#item	#interaction	sparsity
MovieLens 1M	6,040	3,952	1,000,209	95.81%

Table 3: Experimental Results on MovieLens

Model	Recall	NDCG
LightGCN	0.2563	0.1996
LightGCN +RandomAttack	0.2545	0.1992
LightGCN +RandomAttack +ConsisRec	0.2617 +2.11%	0.2043 +2.35%

5.8. Comparison between ConsisRec with Anti-fakeu

You et al. [5] reported the Recall and NDCG of the base recommender and Anti-fakeu under shilling attack scenarios. The results indicate that Anti-fakeu degrades recommendation performance, leading to lower accuracy compared with state-of-the-art methods. In contrast, our method achieves even higher recommendation performance than the recommender trained without shilling attacks, highlighting the ability of our noise-filtering–based approach to improve recommendation accuracy.

6. Conclusion and Future Work

In this paper, we propose ConsisRec, a scalable and model-agnostic post-processing framework for anti-shilling recommendation. Unlike existing anti-shilling methods that are tightly coupled with specific recommendation models or training procedures, ConsisRec operates purely on the output recommendation lists of base recommenders, enabling flexible deployment without modifying model architectures or retraining processes. Preliminary experiments show that ConsisRec improves recommendation accuracy under random shilling attacks, achieving performance gains over the base LightGCN model. There are several promising directions for future work. First, we plan to evaluate ConsisRec on more datasets and additional backbone recommender models, including directly applying ConsisRec to the base recommender model, to further verify its generality. Second, we will extend the evaluation to stronger and more sophisticated shilling attack strategies and conduct systematic comparisons with state-of-the-art anti-shilling defense methods. Third, we will incorporate direct anti-shilling metrics, such as Target Item Exposure (TIE), into comprehensive evaluations to better quantify defensive effectiveness.

7. Acknowledgments

This work is supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant Number 25K21304.

References

[1] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017, April). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182).

[2] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020, July). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval* (pp. 639-648).

[3] Liu, F., Cheng, Z., Zhu, L., Gao, Z., & Nie, L. (2021, April). Interest-aware message-passing GCN for recommendation. In *Proceedings of the web conference 2021* (pp. 1296-1305).

[4] Zhang, S., Yin, H., Chen, T., Hung, Q. V. N., Huang, Z., & Cui, L. (2020, July). Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 689-698).

[5] You, X., Li, C., Ding, D., Zhang, M., Feng, F., Pan, X., & Yang, M. (2023, April). Anti-fakeu: Defending shilling attacks on graph neural network based recommender model. In *Proceedings of the ACM web conference 2023* (pp. 938-948).

[6] Mu, L., Liu, Z., Zhu, Z., & Lin, Z. (2025, April). Trust-GRS: A Trustworthy Training Framework for Graph Neural Network Based Recommender Systems Against Shilling Attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 39, No. 12, pp. 12408-12416).

[7] Tang, J., Wen, H., & Wang, K. (2020, September). Revisiting adversarially learned injection attacks against recommender systems. In Proceedings of the 14th ACM Conference on Recommender Systems (pp. 318-327).

[8] Bourtoule, L., Chandrasekaran, V., Choquette-Choo, C. A., Jia, H., Travers, A., Zhang, B., ... & Papernot, N. (2021, May). Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)* (pp. 141-159). IEEE.

[9] Mo, F., & Yamana, H. (2023). EPT-GCN: Edge propagation-based time-aware graph convolution network for POI recommendation. *Neurocomputing*, *543*, 126272.

[10] Mo, F., Fan, X., Chen, C., & Yamana, H. (2025). Synergistic fusion framework: Integrating training and non-training processes for accelerated graph convolution network-based recommendation. *Pattern Recognition*, 111829.

[11] Huang, H., Mu, J., Gong, N. Z., Li, Q., Liu, B., & Xu, M. (2021). Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644*.

[12] Lam, S. K., & Riedl, J. (2004, May). Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web* (pp. 393-402).

[13] Song, J., Li, Z., Hu, Z., Wu, Y., Li, Z., Li, J., & Gao, J. (2020, April). Poisonrec: an adaptive data poisoning framework for attacking black-box recommender systems. In *2020 IEEE 36th international conference on data engineering*

(ICDE) (pp. 157-168). IEEE.

[14] Wu, F., Gao, M., Yu, J., Wang, Z., Liu, K., & Wang, X. (2021). Ready for emerging threats to recommender systems? A graph convolution-based generative shilling attack. *Information Sciences*, *578*, 683-701.

[15] Zhang, S., Yin, H., Chen, T., Hung, Q. V. N., Huang, Z., & Cui, L. (2020, July). Gcn-based user representation learning for unifying robust recommendation and fraudster detection. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval* (pp. 689-698).

[16] Burke, R., Mobasher, B., Williams, C., & Bhaumik, R. (2006, August). Classification features for attack detection in collaborative recommender systems. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 542-547).

[17] Bhaumik, R., Mobasher, B., & Burke, R. (2011). A clustering approach to unsupervised attack detection in collaborative recommender systems. In *Proceedings of the International Conference on Data Science (ICDATA)* (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

[18] Wu, Z., Wu, J., Cao, J., & Tao, D. (2012, August). HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 985-993).

[19] Zhang, Y., Hao, Q., Zheng, W., & Xiao, Y. (2025). User similarity-based graph convolutional neural network for shilling attack detection. *Applied Intelligence*, *55*(5), 340.

[20] Li, H., Gao, M., Zhou, F., Wang, Y., Fan, Q., & Yang, L. (2021). Fusing hypergraph spectral features for shilling attack detection. *Journal of information security and applications*, *63*, 103051.

[21] Hao, Y., Meng, G., Wang, J., & Zong, C. (2023). A detection method for hybrid attacks in recommender systems. *Information Systems*, *114*, 102154.

[22] Mo, F., Fan, X., Chen, C., Bai, C., & Yamana, H. (2024). Sampling-based epoch differentiation calibrated graph convolution network for point-of-interest recommendation. *Neurocomputing*, *571*, 127140.

[23] Zhou, X., Lin, D., Liu, Y., and Miao, C. 2023. Layer-refined graph convolutional networks for recommendation. In Proceedings of the 2023 IEEE 39th International Conference on Data Engineering, pp. 1247-1259.

[24] Wang, Z., Yu, J., Gao, M., Yuan, W., Ye, G., Sadiq, S. W., & Yin, H. (2024). Poisoning Attacks and Defenses in Recommender Systems: A Survey. *CoRR*.

[25] Wang, X., He, X., Wang, M., Feng, F., & Chua, T. S. (2019, July). Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval* (pp. 165-174).

[26] Lin, Z., Tian, C., Hou, Y., & Zhao, W. X. (2022, April). Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022* (pp. 2320-2329).

[27] Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., & Nguyen, Q. V. H. (2022, July). Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 1294-1303).

[28] Huang, H., Mu, J., Gong, N. Z., Li, Q., Liu, B., & Xu, M. (2021). Data poisoning attacks to deep learning based recommender systems. *arXiv preprint arXiv:2101.02644*.

[29] Zhang, S., Yin, H., Chen, T., Huang, Z., Nguyen, Q. V. H., & Cui, L. (2022, February). Pipattack: Poisoning federated recommender systems for manipulating item promotion. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (pp. 1415-1423).

[30] Mo, F., & Yamana, H. (2022, November). GN-GCN: Combining geographical neighbor concept with graph convolution network for POI recommendation. In *International conference on information integration and web* (pp. 153-165). Cham: Springer Nature Switzerland.

[31] Fang, M., Yang, G., Gong, N. Z., & Liu, J. (2018, December). Poisoning attacks to graph-based recommender systems. In *Proceedings of the 34th annual computer security applications conference* (pp. 381-392).

[32] Chen, C., Mo, F., Fan, X., Bai, C., & Yamana, H. (2023, March). Mobarec-gcnfp: Champion recommendation for multi-player online battle arena games using graph convolution network with fewer parameters. In *2023 IEEE 8th International Conference on Big Data Analytics (ICBDA)* (pp. 147-153). IEEE.

[33] Fan, X., Mo, F., Chen, C., Bai, C., & Yamana, H. (2024, March). Connectivity-Aware Experience Replay for Graph Convolution Network-Based Collaborative Filtering in Incremental Setting. In *2024 9th International Conference on Big Data Analytics (ICBDA)* (pp. 233-242). IEEE.

小規模言語モデルが抽出する経験価値に基づくゲーム推薦システムの構築

長尾　羽留†　　亀谷　由隆††

† 名城大学大学院理工学研究科情報工学専攻　〒 468–0073 名古屋市天白区塩釜口一丁目 501 番地
†† 名城大学情報工学部情報工学科　〒 468–0073 名古屋市天白区塩釜口一丁目 501 番地
E-mail: ††ykameya@meijo-u.ac.jp

あらまし　近年巨大化の一途を辿っているゲーム市場では情報検索の仕組みが必須であるが，従来の検索方法ではゲームの本質を捉えることは困難である．そこで本研究では，ゲームは他のマルチメディアと比較して自らが働きかける側面，すなわちインタラクティブ性が強いということに注目し，プレイヤー自身の経験によって得られる価値（経験価値）が重要であると考える．具体的には，Steam Web API から収集したユーザレビューに対し，小規模言語モデルによって同じ主張をしているレビュー同士をグループ化し，その中で共通する観点を経験価値として抽出する．そして，それらの観点に基づく絞り込み条件をユーザが指定できるゲーム推薦システムを提案する．被験者実験では，好意的な評価も多くあったが観点の分かりやすさや全体的なシステムの使いやすさに関して課題が残った．

キーワード　ゲーム推薦，経験価値，レビュー分析，小規模言語モデル

1　はじめに

近年巨大化の一途を辿っているゲーム市場において，従来のキーワード検索ではゲームの本質を捉えられず，ユーザが求めているゲームを見つけ出すことは困難である．特に Web 検索では，概略の情報や一般的な知識などといった期待しない不要なページを多く含んでしまう．この問題に対処するために本研究では，ゲームは他のマルチメディアと比較して自らが働きかける側面，すなわちインタラクティブ性が強い [12] ということに注目し，ゲームの本質を捉えるには，その商品を購入した人自身の経験によって得られる価値（経験価値）が重要であると考えた．

経験価値とは，1990 年代末にコロンビア・ビジネス・スクールの B. H. Schmitt によって提唱されたマーケティング手法の概念 [9] であり，一般的に 5 つの戦略的経験モジュール (strategic experiential module, SEM) と呼ばれるカテゴリに大きく分類される．その 5 つとは，SENSE（感覚的知覚），FEEL（情緒的感情），THINK（創造的思考），ACT（身体性），RELATE（社会性）である．SENSE は五感，ゲームであれば特に視覚・聴覚の直接的な刺激によって得られる価値である．FEEL は感情の動きによって得られる価値である．これは理性によって制御できない領域である衝動的な感情を扱う．THINK は知的好奇心や創造心の刺激によって得られる価値である．ACT は他者との接触が生じたときや，生理的な欲求，さらにはライフスタイルの変化によって得られる．RELATE では他者との繋がりを求める欲求を満たすことや，単に所属することによる安心によって得られる．

中谷ら [7] は，事前に人手で作成した辞書を用いて経験価値をレビュー文から抽出し，ゲームの経験を定量化し経験値へと変換させ，個々のユーザの経験に適するゲームを推薦するシステムを提案した．その中で，経験価値の取得にユーザレビューを選択した．その理由は，ユーザレビューには “画像がとても汚い” などの主観的な評価や “何度も泣いてしまった” などの感情的な文章が含まれていて，これらはゲームの経験価値を表している，ということである．

本研究では，Steam Web API から収集したユーザレビューを対象に小規模言語モデル (small language model, SLM) Qwen3-8B[1] を用いて，ゲーム体験に関する経験価値の観点を抽出する．提案手法では各レビューから観点を抽出し，同一または類似の主張を持つ観点同士を段階的にグループ化する．このレビュー集約を繰り返すことで詳細な観点グループを徐々に一般化させ，最終的にユーザが選択可能な観点名を出力する．この得られた観点を絞り込み条件を指定できるゲーム推薦システムを提案する．

経験価値を観点として抽出することは，豊田ら [10] の研究から着想を得ている．豊田らは，食べログのレビューから Google Cloud Platform の Natural Language API[2] を用いてエンティティを抽出し，エンティティの分散表現に基づいてクラスタリングしたエンティティ集合から，コサイン類似度に基づき各クラスタの代表語を観点として抽出した．そして，それらの観点を絞り込み条件としてインタフェース表示することで，飲食店の特徴を捉えられる推薦システムを提案した．

また，レビュー集約は中井ら [6] の研究からも着想を得ている．中井らは，大規模言語モデル (large language model, LLM) を用いてレビューから観点と評価を抽出し，類似している観点・評価のクラスタリングを行った．そして観点ごとに評価を対応付けた表を作成することで，人の意見を基に短時間で商品比較可能なシステムを提案した．

1：https://huggingface.co/Qwen/Qwen3-8B
2：https://cloud.google.com/natural-language?hl=ja

本論文の構成を示す．まず第 2 節で先行研究について述べる．次に第 3 節で提案システムの構成や仕様について述べる．第 4 節で実施した被験者実験の方法を述べ，示した結果について考察する．最後に第 5 節で本論文をまとめる．

2 先行研究

ユーザレビューから抽出したアイテムの特徴量を中心とした推薦システムの開発に関しては，以前から様々な研究がなされている．大山ら [8] は，ゲームレビューサイトに掲載されているレビュー文を word2vec[3]に学習させ，言葉の足し引きによってゲームを定量化したゲーム推薦システムの研究を行った．しかし，この大山らの研究，前述の豊田らの研究いずれも経験価値を上手く抽出できていない，単語単位でレビュー分析を行っているために感情極性を考慮できていないなどの課題が挙げられている．

また，近年の深層学習の進展は文章分析や情報抽出 (information extraction, IE) の分野に大きな影響を与えており，非構造化テキストから構造化情報を抽出する上で重要な役割を果たしている．例えば Wu ら [11] は，エンティティの予測集合と正解集合の類似度を測る柔軟な評価指標 AESOP (Approximate Entity Set OverlaP) の提案および，出力トークン数削減と多段階処理による効率性と精度の向上を実現した言語モデル MuSEE (Multi-Stage Structured Entity Extraction) の開発を通して，従来の IE よりも多角的な視点から洞察を得られるような枠組み SEE (Structured Entity Extraction) を提案した．Korikov ら [5] は，ユーザクエリを LLM によって複数の観点に自動的に分解し，レビュー文書を観点ごとに順位付けしたうえで統合する Aspect Fusion を提案した．これにより，従来の単一クエリ処理では捉えにくい多観点クエリをもつ検索タスクにおいて高い再現性と精度を実現した．

3 提案システム

本節では，提案システムについて説明する．まず 3.1 節で提案システム全体の構成について述べる．次に 3.2 節で用いたデータセットについて述べる．さらに 3.3 節でレビュー集約プロセスと各プロセスの結果について述べる．

3.1 システムの構成

後述する 3.3 節で作成した観点データを利用して Web 推薦システムを実装した．実装には Python の Web 開発用フレームワークである Flask [4] を用いた．各ゲームに関する概要や動画などの情報は，レビューと同様に Steam Web API のゲームの詳細情報を取得するリクエスト [13] を用いて取得した．図 1 は実装したシステムの構成図である．

図 1 の通り，Steam Web API からのレビュー取得およびレビュー集約はシステムを利用するための前処理段階であり，レビュー分析結果を Flask アプリケーションに保存してからユーザが使用できる状態になる．図 2, 3 はそれぞれユーザ側から確認できる検索画面と検索結果画面である．

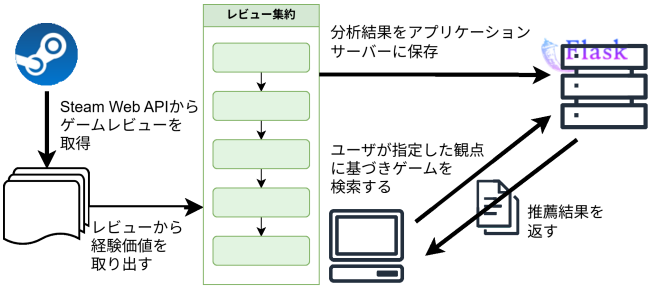

図 1 提案システムの構成図

検索画面は図 2 の通り，左側の観点エリアと右上の選択観点表示エリア，右下の説明文エリアに分かれる．観点エリアでは抽出された観点が 1 節で述べた SEM およびサブグループごとに表示されている．サブグループは折り畳みのユーザインタフェース（以下，単に UI）になっており，類似評価グループは絞り込み条件としてチェックボックスで選択できる．ここで，サブグループおよび類似評価グループは中井らの先行研究 [6] に基づいて設定したグループ階層であり，SEM→ サブグループ → 類似評価グループの階層関係で構成されている．この詳細については 3.3 節で述べる．ユーザの選択した観点が右上のエリアに表示されている状態で「観点で検索」ボタンを押すことでゲーム推薦を受けられる．また，サブグループ・観点にマウスを合わせると右下のエリアに説明文が表示される．

検索画面は豊田ら [10] の研究に基づいてレイアウトを設計した．豊田らは実装した飲食店推薦システムの中で，レビューから抽出した観点を「利用シーン」「味」「料理名」「お店の特徴・雰囲気」「その他」の 5 つのカテゴリに分類し，各カテゴリ内の観点をチェックボックスで選択できるような UI を実現している．

検索結果画面は図 3 の通り，左側の推薦ゲームリストエリアと右側のゲーム詳細情報エリアに分かれている．ゲームリストエリアでは，各ゲームのタイトルとヘッダー画像，選択した観点のうち一致するものの一部を 1 つのパネルに表示する．このパネルのいずれかをクリックすると詳細情報エリアが書き換わる．詳細情報エリアでは，ゲームの説明や動画・スクリーンショット，ユーザが選択した観点のうち一致したものを表示する．

3.2 データセット

本研究では，Steam が公式で提供している Steam Web API のレビュー情報取得 URL[4]にアクセスして得られたユーザレビューを用いた．しかし，現在 Steam で扱っているゲームを全て用いるのは規模が大きすぎ，また馴染みのないゲームタイトルは評価が難しいため，今回は AAA タイトルを輩出している企業や日本の有名企業から発売しているゲームタイトルの中で，レビュー数が 10 以上あるアイテムのみに絞り込みを行った．そ

3：https://radimrehurek.com/gensim/models/word2vec.html

4：https://partner.steamgames.com/doc/store/getreviews

図 2　検索画面のレイアウト

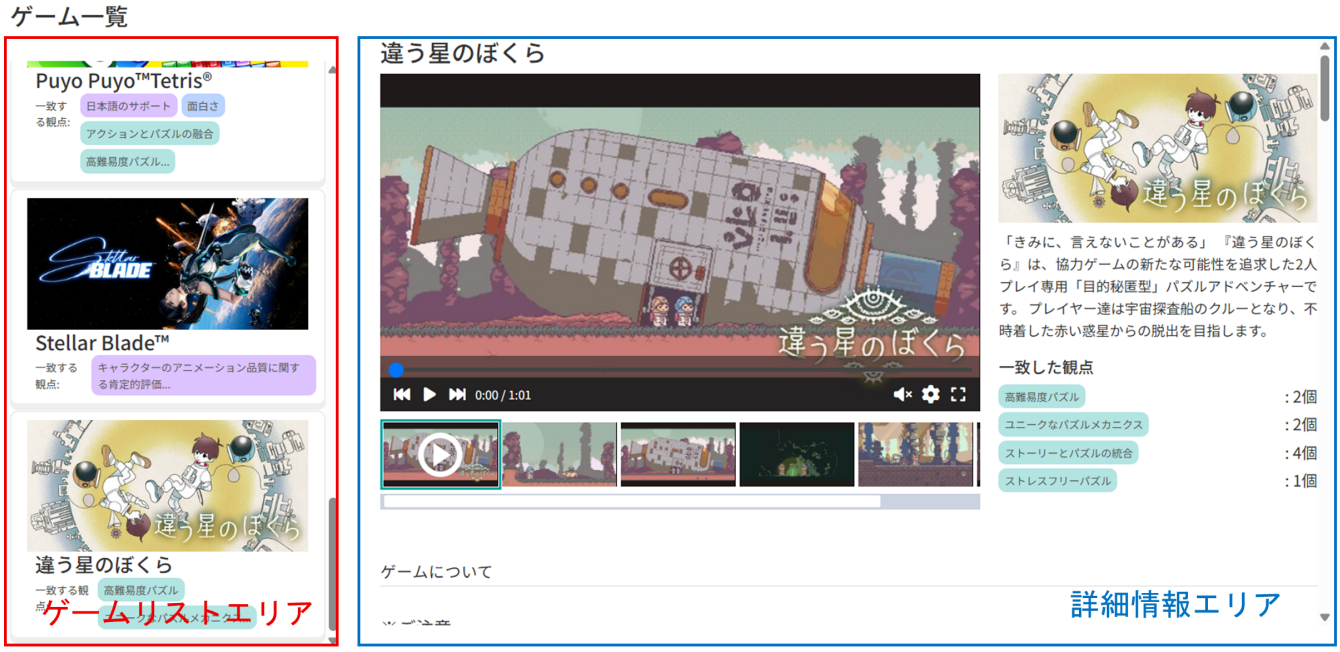

図 3　検索結果画面のレイアウト（画像の出展：Steam）

の結果，タイトル数は 1208，全レビュー数は 9196 となった．

3.3　レビュー集約処理

本節では，収集したレビューデータから観点を抽出し提案システムに適した形に加工する処理（図 1 の左側）について説明する．これは提案システムが使用可能になるまでの前処理である．ここでは中井らの先行研究 [6] を参考に，図 4 に表される 5 プロセスからなる処理を行った．5 つのプロセスはいずれでも SLM (Qwen3-8B) を利用している．

3.3.1　観点と評価を抽出と SEM への分類

3.2 節で収集したレビューを 1 つずつ Qwen3-8B に入力し経験値を観点と評価という形で抽出した．評価はゲームの経験値を表す部分をレビューから抜き出した 1 文であり，観点は評価を端的に表す名前である．図 5 は，以上の処理の一例である．

3.3.2　観点のサブグループへの集約

プロセス 3.3.1 で抽出した観点を Qwen3-8B に入力し，似ている主張をしているもの同士を 1 つのサブグループに集約する．処理のフローは図 6 の通りであり，階層クラスタリングに似た処理を行う．

具体的には，観点リストを固定長（今回は 25 個）の小集合に区切り，各小集合から Qwen3-8B を用いたクラスタリングによって生成したグループ名を取り出し，1 つのグループ名リストを作る．このリストをまた固定長に区切り，前工程と同様にクラスタリングするという処理を繰り返し，徐々に一般的なグループ名にしていく．クラスタリング結果に変化が見られなくなった段階で得られたグループ名を最終的なサブグループと定

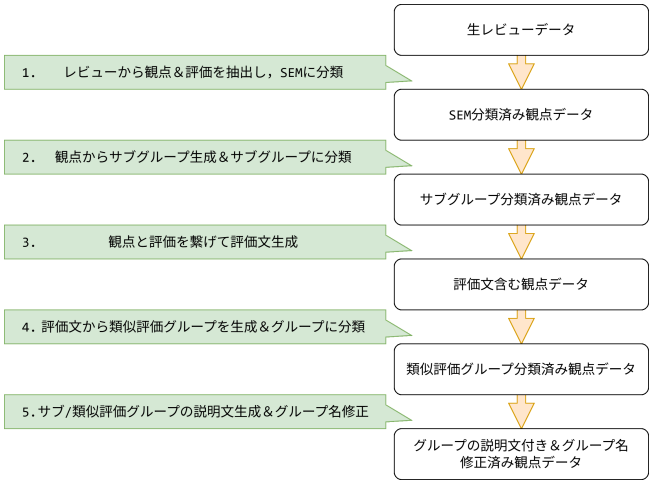

図 4　レビュー集約処理の全体図

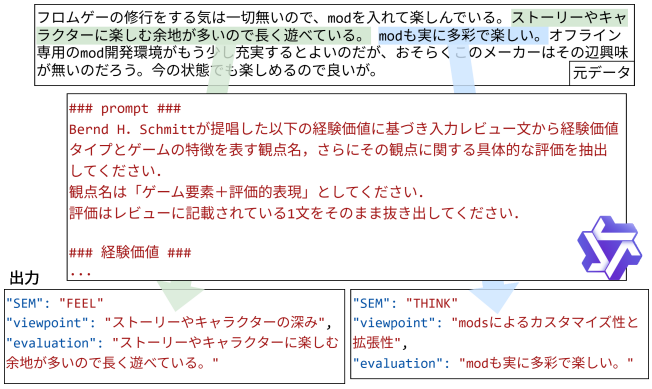

図 5　レビューから観点&評価抽出の例

義する．その後，観点を得られたサブグループに分類していく．図 7 は以上の処理の具体例である．

このような方法を取ったのは，Qwen3-8B のコンテキスト長上限の中で効果的にレビュー集約するためである．9000 件以上のレビューを扱っているとプロンプトがコンテキスト長上限を超えてしまうことが避けられない．Mamba [2] などコンテキスト長上限が高いモデルもあるが，プロンプトが長すぎても抽出精度向上は見込めないと考えた．また，Qwen3 の思考モードを効果的に活用するには，クラスタリングは段階的に行うべきであると考えた．この点は，比較的少数のレビューを扱っていたと思われる中井ら [6] の研究では見られなかった視点であると考えている．

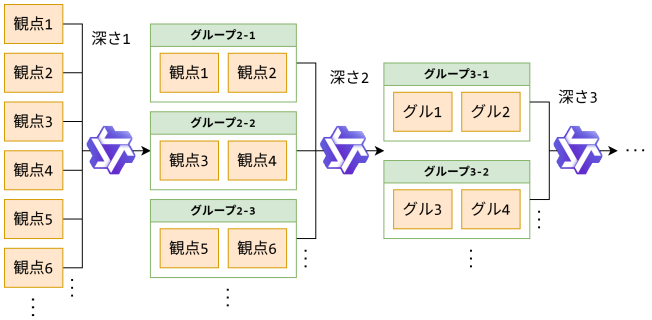

図 6　観点のサブグループへの集約フロー

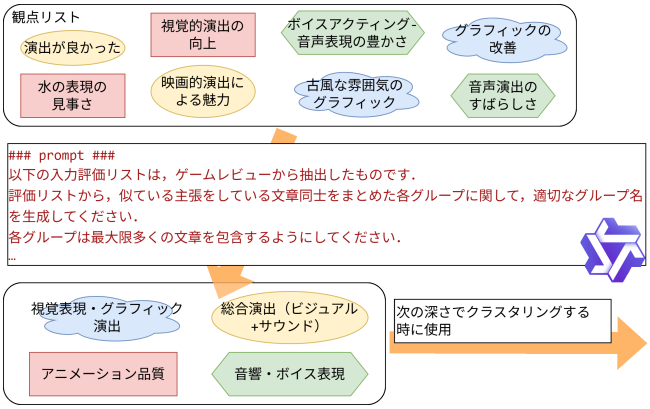

図 7　グループ名生成の具体例

3.3.3　評価文の生成

3.3.1 節で抽出した観点と評価を繋げ，自然な日本語文を生成する．3.3.2 節で得られたサブグループごとに類似評価の集約を行う．評価集約では，各観点グループの中にどのような意見があるのかを簡潔に示すために似ている評価を 1 文に要約する．しかし，評価のみを使用して要約すると情報が不足する場合がある．例えば，「グラフィック」グループのある観点に対して「いいもの」という評価が抽出された場合，「映像美」に対する「いいもの」という評価と「アートスタイル」に対する「いいもの」という評価は 1 つにまとめるべきではない [6]．そこで評価を集約するために，観点と評価を 1 文にまとめた擬似的なレビュー文を作成した．図 8 は，以上の処理の一例を表している．

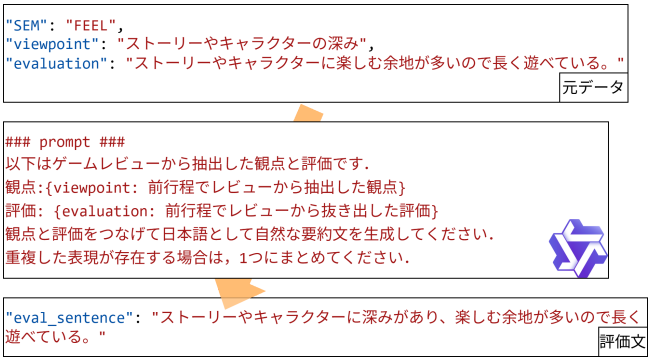

図 8　評価文生成の具体例

3.3.4　評価文の類似評価グループへの集約

3.3.3 節で生成した評価文を利用して，3.3.2 節と同様に同じ主張をしている評価文を集約し，類似評価グループ名を取得する．その後，類似評価グループに分類する．

3.3.5　グループの説明文の生成とグループ名の修正

サブグループ，類似評価グループについて説明文を付与することで，どのような観点がゲームに存在するのかユーザが把握するのを補助する．その後，得られた説明文に基づいてグループ名をよりユーザに分かりやすいものに修正する．

4 被験者実験

本節では提案システムを用いた被験者実験の内容と結果，それに対する考察を述べる．情報系の学生 20 名（修士 2 年: 3 人, 学部 4 年: 8 人, 学部 3 年: 9 人）の協力を得て提案システムを用いた実験を行い，結果を記録した．

被験者実験内容は以下の 2 つである．

ゲーム推薦の評価実験（以下，評価実験 A）

提案システムを用いて，3 つのシナリオに沿ったゲームを探索する．各シナリオ探索終了後に提案システムによるゲーム推薦を評価する質問に回答する．

UI・ユーザビリティの評価実験（以下，評価実験 B）

提案システムのインタフェースおよびユーザビリティに関する質問に回答する．

4.1 ゲーム推薦の評価実験

4.1.1 実 験 概 要

評価実験 A において用いるシナリオは表 1 の 3 つである．被験者は各シナリオに沿って提案システムを用いて遊びたいと感じるゲームを探索する． 被験者は各シナリオに沿ったゲーム探索がすべて終了した後に表 2, 3 の 5 段階リッカート尺度における評価の質問に回答する．この 2 つの表は提案システムにて用いた観点を使った推薦において，それぞれ推薦条件と推薦結果に関する評価を尋ねるものである．

表 1〜3 の項目は豊田ら [10] の研究に基づいて設定した．

表 1　3 つのゲーム探索シナリオ

No.	シナリオの内容
1	あなたは，美しいビジュアルとサウンドの融合を楽しみつつ，物語に没入できるゲームをプレイしたいと考えています．その条件に当てはまるようなゲームを 1 つ探してください．
2	あなたは，ビジュアルとサウンド，物語の没入性のレベルが高く，多プレイヤーと協力・交流ができるゲームで，マルチプレイでがっつりやり込みたいと考えています．その条件に当てはまるようなゲームを 1 つ探してください．
3	あなたは，ストーリーとキャラクターが丁寧に描かれつつ，挑戦を通じた発見や交流があり，かつコストパフォーマンスが高いゲームをプレイしたいと考えています．その条件に当てはまるようなゲームを 1 つ探してください．

表 2　評価実験 A 推薦条件に関する質問項目

No.	質問内容
1	観点は役に立ったか
2	観点は分かりやすかったか
3	特徴的な観点を発見できたか
4	選択した観点と推薦されたゲームは関連があったか
5	選択した観点はそのゲームの特徴を反映できていると感じたか

各シナリオの表 2, 3 の回答は，ノンパラメトリック検定の一つである一標本の Wilcoxon 符号付き順位検定 (one-sample Wilcoxon signed rank test) を用いた（以下，単に Wilcoxon 符号順位検定と呼ぶ）．

表 3　評価実験 A 推薦結果に関する質問項目

No.	質問内容
1	推薦されたゲームを実際に遊びたいと感じたか
2	推薦されたゲームは気に入ったか
3	推薦されたゲームは見つけにくい良作と感じたか
4	詳細情報画面からそのゲームの特徴が分かったか
5	選択した観点とレビュー文内の該当箇所は関連していたか

帰無仮説 (H_0)，対立仮説 (H_1) を次のように立てた．

帰無仮説 (H_0)

母集団の中央値はリッカート尺度における中央の値 3 と等しい．

対立仮説 (H_1)

母集団の中央値はリッカート尺度における中央の値 3 よりも大きい．

本検定法では，標本の各データと標本の代表値とリッカート尺度における中央の値 $\mu_0 = 3$ の差から統計量 T を求める．この T が Wilcoxon 符号順位検定数表 [14] における棄却限界値以下のとき帰無仮説を棄却する．すなわち，母集団の中央値は μ_0 よりも大きいと結論する．標本サイズが 25 を超える場合は T を用いて別の統計量を求めるが，本研究で扱う標本サイズは 25 以下のため上記の判断を行った．

4.1.2 実 験 結 果

表 4, 5 は評価実験 A の各シナリオの結果を示している．表中の * がついている T は帰無仮説を棄却することを示している．

表 4　評価実験 A 表 2 の質問の回答結果

No.	シナリオ 1		シナリオ 2		シナリオ 3	
	棄却域	T	棄却域	T	棄却域	T
1	$T \leqq 47$	*0.0	$T \leqq 47$	*0.0	$T \leqq 47$	*4.0
2	$T \leqq 35$	36.0	$T \leqq 35$	*24.5	$T \leqq 35$	*13.5
3	$T \leqq 47$	*6.0	$T \leqq 53$	*0.0	$T \leqq 53$	*19.5
4	$T \leqq 47$	*0.0	$T \leqq 53$	*4.5	$T \leqq 53$	*10.0
5	$T \leqq 60$	*0.0	$T \leqq 41$	*0.0	$T \leqq 41$	*10.0

表 5　評価実験 A 表 3 の質問の回答結果

No.	シナリオ 1		シナリオ 2		シナリオ 3	
	棄却域	T	棄却域	T	棄却域	T
1	$T \leqq 35$	*30.0	$T \leqq 47$	*18.0	$T \leqq 47$	*26.0
2	$T \leqq 30$	37.5	$T \leqq 41$	*23.5	$T \leqq 35$	*16.5
3	$T \leqq 53$	*6.5	$T \leqq 21$	24.0	$T \leqq 30$	*11.0
4	$T \leqq 47$	*2.0	$T \leqq 60$	*4.5	$T \leqq 53$	*0.0
5	$T \leqq 47$	*6.5	$T \leqq 53$	*3.5	$T \leqq 47$	*4.5

表 4, 5 より，シナリオ 1 では「観点は分かりやすかったか」「推薦ゲームは気に入ったか」の質問項目でリッカート尺度の

中央の値 $\mu_0 = 3$ とは有意差が見られず，それ以外の質問項目では有意差が見られた．シナリオ 2 では「推薦されたゲームは見つけにくい良作だと感じたか」の質問項目で有意差が見られず，それ以外の質問項目で有意差が見られた．シナリオ 3 では全ての質問項目において有意差が見られた．

4.1.3 考　　察

シナリオ 1 では「観点は分かりやすかったか」「推薦されたゲームは気に入ったか」の 2 項目で有意差を示すことができなかった．被験者から感じたことや意見を自由記述として得られたので，その中からどのようなことが課題点となるのかを考察する．

「観点は分かりやすかったか」という質問項目と関連がある意見として

- 「多面的な没入体験」の「質の鑑賞」を中心に検索しましたが，ぱっと見たときに五感のどの部分と関連があるのかが分かりづらいかなと感じました．
- 「物語への没入」について検索しようとしたが,「物語と没入体験」,「物語表現・没入要素」や「没入型物語」など同じような言葉の観点が多く，違いがわからなかった．

というものが得られた．これはレビューから抽出した観点に不適切なものが混ざってしまい，結果として分かりづらい名前になってしまったのだと考えられる．その原因として，Qwen3-8B に経験価値について十分な情報を持たせられなかった点や，同じような主張をしている観点の集約が不足している点が考えられる．

「推薦されたゲームは気に入ったか」に関しては

- 推薦されたゲームがどのようなものかぱっと見でわかるとさらに良いと感じた．例えば，ゲームジャンルなどは別枠で表示したりすると，どんなゲームかが想像しやすい．
- 自分が選んだゲームが動画が反映されていないゲームであったため，ゲームの操作感やシナリオにある実際の音声・サウンドを評価できないことに残念に思った．

など，推薦されたゲームの見せ方に関して否定的な意見が見られた．また

- 選択する観点が多いと感じた．そのため，選択すべき観点を探すことに少し手間取ってしまい，使いづらさを感じた．
- 観点が多く，探しにくい

など，観点での検索に関しても否定的な意見が見られた．

シナリオ 2 においても「観点は分かりやすかったか」「推薦されたゲームは気に入ったか」の 2 項目は有意差は見られたものの，棄却域と検定統計量 T との差が他の質問項目と比較して小さい結果になっている．

シナリオ 3 では「推薦されたゲームを実際にプレイしたいと感じたか」の質問項目が，棄却域と T との差が他の質問項目と比較して小さい結果になった．この項目に関して低く評価した被験者の 1 つの意見として

- 「ストーリーとキャラクター」という観点を選択したところ，レビュー文では批判的な意見での観点抽出が行われていたため，プラス面でのゲーム推薦が行われると考えていた自分としてはギャップに感じた．

という記述があった．実際に今回は感情極性を考慮せずに観点抽出を行ったので，ネガティブな意見がノイズになってしまったのだと考えられる．

以上の 3 つのシナリオの結果を総合すると，観点抽出におけるポジティブ・ネガティブのフィルタリング，経験価値を踏まえた SLM のプロンプティングやファインチューニング，検索・検索結果画面での大まかな情報と詳細情報の分別と，それらの効果的な見せ方をよく検討する必要がある．

4.2 評価実験 B

4.2.1 実 験 概 要

一般的に UI はシステムを運用することで気づく不確定要素が多く，これには UI の重要性に気付くことが難しい点が関係している可能性がある [3]．そのため，評価実験 B では被験者は提案システムのインタフェースおよびそのユーザビリティを評価するため，表 6, 7 の質問に回答する．質問への回答は 5 段階リッカート尺度における評価である．

表 6, 7 の項目も，評価実験 A と同様に豊田ら [10] の研究に基づいて設定した．

表 6　評価実験 B UI の質問項目

No.	質問内容
1	文字の大きさは適切だったか
2	レイアウトは適切だったか
3	観点の選択や追加はしやすいと感じたか
4	選択観点エリアのサブグループのカテゴリは適切だったか
5	推薦結果一覧は，ゲームの概要を素早く把握できるスタイルで構成されていたか
6	詳細情報画面は，ゲーム推薦における重大な情報が表示されていたか

表 7　System Usability Scale (SUS) 質問項目

No.	質問内容
1	頻繁に使用したいと思う
2	不必要に複雑だった
3	使いやすいと感じた
4	利用するには技術者のサポートが必要だと思う
5	機能はまとまっていると思う
6	矛盾がとても多いと感じた
7	ほとんどの人がすぐに使いこなせると思う
8	使うのが面倒に感じる
9	自信をもって操作できた
10	使いこなすにはたくさんの事前知識が必要だと思う

表 6 の回答は，評価実験 A と同様に Wilcoxon 符号順位検定を用いて有意差検定を行った．表 7 の回答は，System Usability Scale (SUS) [1] を用いて評価を行った．SUS はそのシステムがユーザにとって使いやすかったかを評価するスコアであり，スコアが高いとユーザビリティが高い．平均の SUS スコアは 68 であり，この値を上回ると一般的に使いやすいシステムといえる．

4.2.2 実験結果

評価実験 B で行った UI 評価の結果は表 8 の通りである．表 8 より，すべての質問項目においてリッカート尺度における中央の値 $\mu_0 = 3$ との有意差が確認できた．

表 8　評価実験 B UI 評価の結果

質問内容	棄却域	T
文字の大きさは適切だったか	$T \leqq 60$	*2.5
レイアウトは適切だったか	$T \leqq 53$	*9.0
観点の選択や追加はしやすいと感じたか	$T \leqq 41$	*20.0
選択観点エリアのサブグループのカテゴリは適切だったか	$T \leqq 35$	*12.0
推薦結果一覧は，ゲームの概要を素早く把握できるスタイルで構成されていたか	$T \leqq 60$	*12.0
詳細情報画面は，ゲーム推薦における重大な情報が表示されていたか	$T \leqq 60$	*5.0

また，SUS の評価実験の結果を表 9 にまとめた．太字の値は SUS の平均といわれている 68 点よりもスコアが上回っていることを示す．表 9 より，各被験者において平均を上回っているスコアは見られたが，全体的なスコア（被験者平均）は平均を下回ったので，本システムは使いやすいとは言えない．

表 9　評価実験 B SUS の結果

被験者 No.	SUS スコア	被験者 No.	SUS スコア
1	**70.0**	11	**75.0**
2	50.0	12	47.5
3	**70.0**	13	35.0
4	**90.0**	14	**87.5**
5	67.5	15	**82.5**
6	**87.5**	16	60.0
7	40.0	17	52.5
8	50.0	18	25.0
9	**80.0**	19	67.5
10	**77.5**	20	45.0
被験者平均			63.0

4.2.3 考　　察

リッカート尺度における中央の値 $\mu_0 = 3$ との有意差は示せたが，棄却域と T との差が小さい質問項目として

- 観点の選択や追加はしやすいと感じたか
- 選択観点エリアのサブグループのカテゴリは適切だったか

が挙げられる．これらに関連する意見として

- 観点が多く，すべてそれなりの長さの日本語で表示されているので，読むのがおっくうでした
- ゲーム検索を行う際，少し項目がどこにあるのか分かりづらいように感じた
- 検索画面において，折り畳みボックスとその中身の整合性をより高めた方が良いと感じた

が見られた．これは，4.1 節と同様に観点抽出の段階で不適切な点があったことがうかがえる．UI の面では，観点の概要を素早く把握させられるレイアウトや，観点項目を分かりやすくするようなレイアウトを検討する必要があると考える．

ユーザビリティ面は特に UI やシステム機能に関する意見が多く見られた．例として

- 更新以外で一度チェックしたものを一括でリセットする機能があるといいなと思いました
- ヘッダーのログアウトの位置が検索画面とゲーム一覧画面で異なるので，ログアウトを一番右に移動するとわかりやすい

などが挙げられる．これらの意見に基づいてシステムを改善して評価テストを繰り返し行うことで，ユーザビリティが向上すると期待される．

5 おわりに

本研究では，SLM を用いてレビューから抽出した経験価値を利用したゲーム推薦システムを構築した．ユーザレビューデータや各ゲームの動画・紹介テキストなどのシステム実装に必要なデータは，Steam Web API のデータ取得 URL にアクセスして取得した．経験価値抽出は複数のレビュー集約プロセスによって構成されている．その中で観点をサブ/類似評価グループに集約するプロセスでは，観点リストを固定長に区切った小集合内でクラスタリングし，得られたグループ名リストをまた同様に処理するという，階層型クラスタリングに着想を得た独自の方法で処理を行った．以上の処理によって得られたデータを用いて Flask によって Web システムを実装した．このシステムを用いて被験者実験を行い，経験価値に基づく観点とシステムの有効性を評価した．

実験の結果，観点に関しては多くの項目で有効性を示すことができたが，主に観点の分かりやすさについて課題が残った．UI に関しても一定の効果を示すことができたが，観点の見せ方について課題が残った．ユーザビリティに関しては残念ながら SUS の平均スコアを上回ることができず，使いやすいシステムであることを示すことができなかった．

今後の課題としては，ポジティブな意見のレビューのみフィルタリング，SLM のプロンプティングの改善やファインチューニングが挙げられる．これらは経験価値抽出の精度向上に必要な改善点である．UI やユーザビリティは，被験者実験アンケートで得られた自由記述意見に基づいて改善していくことが必要である．

また，本研究では対象領域をゲームに限定して分析を行ったが，本研究で扱った「経験価値」はマーケティング分野において広く用いられている概念であり，特定のコンテンツ領域に依存しない汎用性を有すると考えられる．このため，本研究で提案した枠組みは音楽や旅行などの他の体験型コンテンツにも応用可能であると期待される．対象領域の拡張を通じて，本手法の有効性および汎用性をさらに検証し，より汎用的かつスケーラブルな分析・推薦手法へと発展させることが望まれる．

文　　献

[1] J. Brooke: SUS: A Quick and Dirty Usability Scale. Usability Evaluation in Industry, Part 6, Chapter 21, pp. 189–194,

1996.
[2] A. Gu and T. Dao: Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv preprint arXiv:2312.00752, 2024.
[3] 風間正弘, 飯塚洸二郎, 松村優也: 推薦システム実践入門 ―仕事でつかえる導入ガイド. オライリー・ジャパン, 2022.
[4] 樹下雅章: Flask 本格入門 ～やさしくわかる Web アプリ開発～. 技術評論社, 2023.
[5] A. Korikov, G. Saad, E. Baron, M. Khan, M. Shah and S. Sanner: Multi-Aspect Reviewed-Item Retrieval via LLM Query Decomposition and Aspect Fusion. SIGIR'24 Workshop on Information Retrieval's Role in RAG Systems, 2024.
[6] 中井香那子, 山本岳洋, 大島裕明: 大規模言語モデルを用いた商品比較のためのレビュー集約. 第 17 回データ工学と情報マネジメントに関するフォーラム (DEIM 2025), 3A-03, 2025.
[7] 中谷知博, 星野准一: 経験価値の分類に基づくゲーム推薦システム. 社会法人 情報処理学会 研究報告, 2008-EC-11, Vol. 2008, No. 129, pp. 49-56, 2008.
[8] 大山浩暉, 竹川佳成, 平田圭二: レビュー文を考慮したゲーム推薦システムの実現に向けた単語の類似度調整の取り組み. エンタテインメントコンピューティングシンポジウム (EC2017), 2017.
[9] B. H. Schmitt: Experiential Marketing. Journal of Marketing Management, Vol. 15, No. 1, 1999.
[10] 豊田陽大, 上野史, 太田学: レビューから抽出した観点を用いた飲食店推薦システムの提案. 第 17 回データ工学と情報マネジメントに関するフォーラム (DEIM 2025), 9E-03, 2025.
[11] H. Wu, Y. Yuan, L. Mikaelyan, A. Meulemans, X. Liu, J. Hensman and B. Mitra: Learning to Extract Structured Entities Using Language Models. The 2024 Conference on Empirical Methods in Natural Language Processing, pp. 6817–6834, 2024.
[12] 4Gamer.net: 【島国大和】ゲームと他のメディアは何がどうちがうんだー！というお話. https://www.4gamer.net/games/000/G000000/20100604072/.
[13] はてなブログ: SteamAPI に軽く触れた記憶. https://271108.hatenablog.com/entry/2023/02/08/204104.
[14] 統計検定のまとめブログ: ウィルコクソンの符号順位検定. https://data-science.gr.jp/theory/tst_wilcoxon_signed_rank_test.html.

一般発表 | Track 3: 情報検索・情報推薦・ソーシャルメディア

2026年3月1日(日) 15:30 ~ 17:40 | G会場

[6G] 推薦システム(最適化/ロバスト性/実運用)

座長:山口 実靖(工学院大学)　コメンテータ:三林 亮太(神戸大学)

17:10 ~ 17:35

[6G-05] [技術報告] DMM.com における検索・レコメンドの取り組み

*森 雄一郎[1]、田中 久温[1] (1. 合同会社DMM.com)

発表者区分：スポンサー
種別：技術報告
インタラクティブ発表：あり

キーワード：推薦タスク、Beyond Accuracy、検索モデル、MLOps

DMMにおける機械学習を活用した検索やレコメンドの取り組みについて、具体的な事例を交えて紹介します。
データドリブンな施策の立案からリリースまでのサイクルにMLOpsを取り入れ、機械学習エンジニアがインフラを意識せず高速に仮説検証を回せるようになった事例についても詳しく触れます。
さらに今回は、レコメンドチームによる「品質を一定に保つ取り組み」や、検索チームによる「ベクトル検索を用いたパーソナライズ強化」といった最新の事例についてもご紹介する予定です。