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

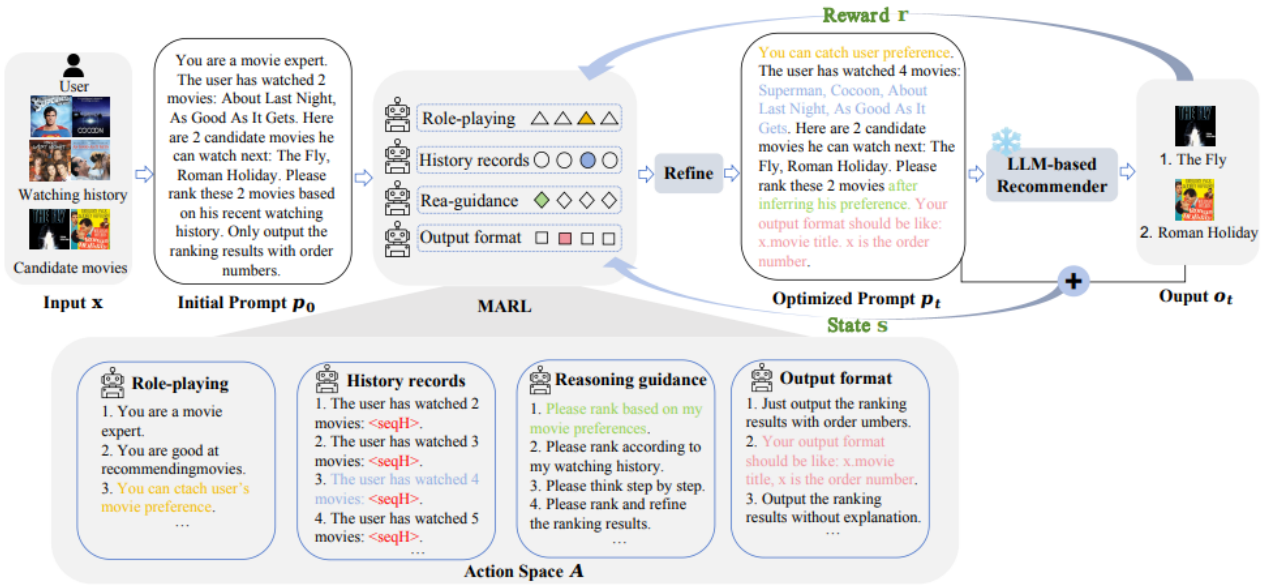

図 1　RPP の概要 [3]

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

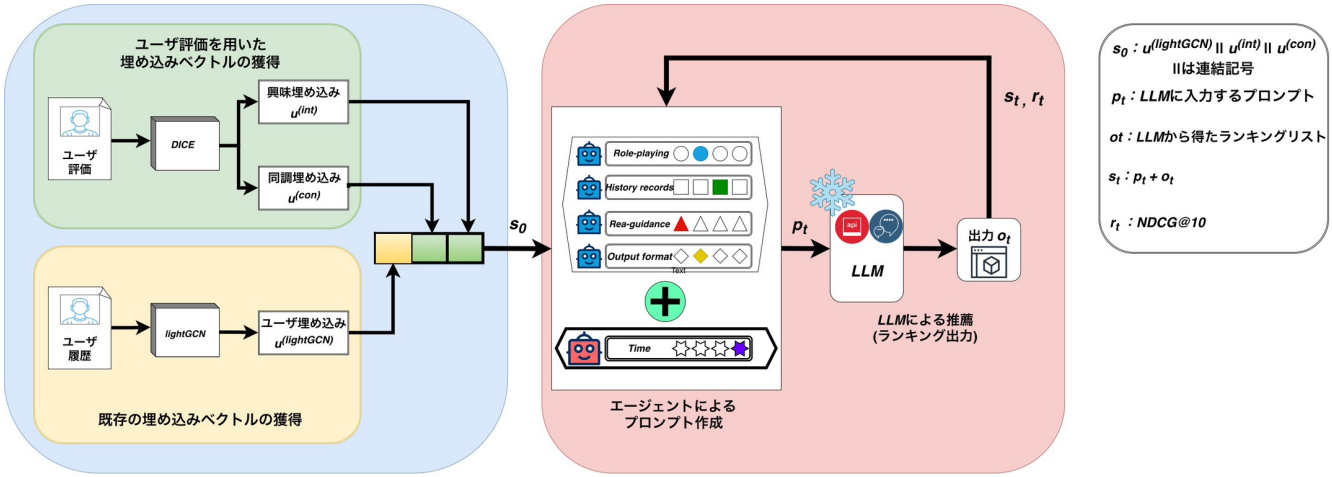

図 2　提案手法の概要

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．

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

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

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

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

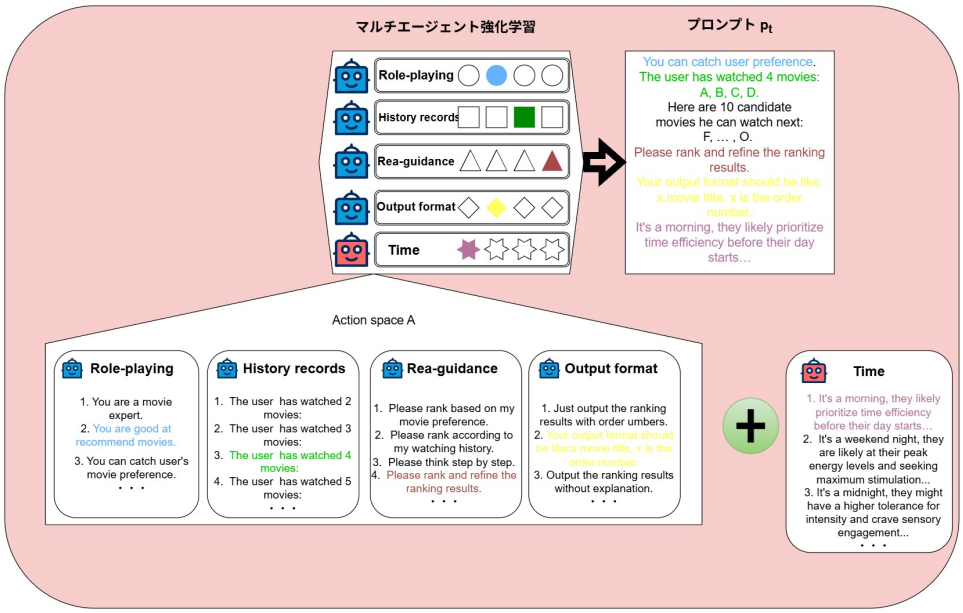

図 3　時間的文脈に基づく行動空間の拡張

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).