個人レポート

グループ: 06 学籍番号: 35714121 氏名: 福富隆大

自己採点

A

理由

強化学習スケジューラーの設計・実装からシミュレーション環境の構築、評価・可視化まで幅広く担当した。単に動作するだけでなく、状態空間の離散化設計や報酬関数の調整など、機械学習として本質的な部分に深く関与した。また、実験結果が仮説を支持しないという想定外の結論に対しても、原因を分析し考察にまとめることができた。仮説検証の姿勢と実装の幅広さの観点からA評価が妥当と判断する。

グループで実装したシステムの説明

システム概要

「ToDoリストの優先順位における強化学習と締切優先等の比較」をテーマとした実験システムである。ユーザーのタスク実行順序について、期限順・重要度順といった単純なルールベース手法と、状況適応型の強化学習（Q-learning）を比較検証する。

誰がどんな時に使うと嬉しいか

多数のタスクを抱える学生・社会人が「どの順番でタスクをこなすべきか」に迷う場面で役立つシステムである。特に以下のような状況で有用性が高い。

- 締切が異なる複数タスクを同時に抱えているとき
- 疲れやすい・集中力が持続しないなど個人差があるとき
- 重要タスクを先にこなすか、締切が近いタスクを優先するかで悩むとき

単純なルール（「締切順」「重要度順」）では考慮されない「その時の集中力」「タスクの種類による疲労の違い」を踏まえた上で最適な順序を提示できる点が特徴である。

システム構成

コンポーネント	内容
タスク管理	60個のタスク、3段階の重要度、期限2〜7日
集中力モデル	疲労蓄積・休憩回復・重要度による疲労増加を再現
4種のスケジューラー	期限順・重要度順・ランダム・強化学習(Q-learning)
シミュレーション環境	7日間・1日8時間の作業時間を模擬
評価・可視化	50回の試行平均スコア・完了率・統計的有意差検定

自分の担当部分の説明

担当範囲

- 強化学習スケジューラーの設計・実装および事前学習
- シミュレーション環境・集中力モデルの実装
- 評価・可視化の実装

---

1. 強化学習スケジューラー

設計方針

タスクを「どれを選ぶか」と直接対応させると行動空間が爆発する（最大60通り）。そこでポリシーベース設計を採用し、行動を「どのポリシーでタスクを選ぶか」の7択に抽象化した。

アクション番号	ポリシー
0	最も重要度が高いタスクを選ぶ
1	締切が最も近いタスクを選ぶ
2	最短時間で終わるタスクを選ぶ
3	スコアが最も高いタスクを選ぶ
4	重要度と締切のバランスで選ぶ
5	集中力に合った難易度のタスクを選ぶ
6	成功確率の高い重要タスクを選ぶ

状態空間の設計

Q-tableのサイズを抑えるため、状態を8次元の離散値で表現した。

次元	内容	区間数
タスク残数	残りタスク数を15個単位で区分	5
高重要度比率	HIGHタスクの割合を3区分	3
締切緊急度	最も近い締切を24時間単位で区分	5
平均所要時間	タスクの平均時間を60分単位で区分	4
集中力レベル	現在の集中力を3段階に離散化	3
疲労蓄積度	疲労の蓄積度を3段階に離散化	3
直前タスク優先度	直前に実行したタスクの重要度	4
直前タスクジャンル	直前のタスクのジャンル	5

状態空間が大きくなりすぎると学習効率が落ちるため、各次元の区間数を絞って探索を効率化した。

報酬関数の設計

報酬は以下の要素を組み合わせて計算する。

```
報酬 = タスクスコア(重要度×所要時間)
     + 完了ボーナス(+200)
     + 締切遵守ボーナス(+100) または 締切違反ペナルティ(−500)
     + 高集中完了ボーナス(+50)
     + ジャンル継続/切り替えボーナス(±30)
     − 連続高優先度ペナルティ(HIGHを連続2回以上で−50×回数)
```

工夫した点として、連続して高重要度タスクを選ぶと疲労が蓄積して効率が落ちることを反映するため、連続高優先度ペナルティを導入した。また、低優先度タスクも完了率向上に貢献するため、LOWタスク完了にも追加ボーナスを設定した。

学習設定

- 学習率: 0.05（安定化のため小さめに設定）
- 割引率: 0.95（長期的な計画を重視）
- ε-greedy: 学習時0.5~0.05 → 評価時0.0（学習済みQ-tableを活用）
- ε減衰率: 0.9998（20,000エピソードで最小値0.05まで緩やかに減衰）

---

2. シミュレーション環境・集中力モデル

シミュレーション環境

シミュレーション全体を管理するクラスを実装し、毎日9:00開始・8時間作業の7日間を模擬した。以下の流れで1ステップを処理する。

1. スケジューラーが次のタスクを選択
2. 残り時間に収まるか安全マージン（85%）で判定
3. 収まらなければ休憩またはその日の作業終了
4. タスク実行 → 集中力更新 → ログ記録

工夫した点として、安全マージン係数（85%）を設けることで、残り時間ギリギリのタスクを選んで途中で日付をまたぐ問題を防いだ。また、タスクリストをディープコピーして各スケジューラーが同一条件で比較できるようにした。

集中力モデル

作業時間に応じて集中力が低下し、休憩で回復するモデルを実装した。

条件	効率倍率	実際の所要時間
高集中（レベル > 0.7）	0.6倍	短縮
低集中	1.4倍	延長

工夫した点として以下を実装した。

- 重要度が高いタスクほど疲れやすい設計（HIGH: 1.6倍の疲労係数）
- 同じ高優先度タスクを連続でこなすとさらに疲労が蓄積（連続ペナルティ1.3倍）
- ユーザーのジャンル好向性タイプに基づき、同じジャンルを続けるか切り替えるかで集中力への影響が変わる

---

3. 評価・可視化

50回の試行を実行し、各スケジューラーの平均スコア・完了率・締切遵守率を集計する評価モジュールを実装した。Welchのt検定による統計的有意差検定も実装し、スケジューラー間の差が偶然ではないかを確認できるようにした。

考察

実験結果

スケジューラー	平均スコア	完了率	統計的有意差
期限順	4802.82 ± 296.71	83.9%	ランダム・RLと同等
重要度順	5037.80 ± 331.22	59.0%	他3手法と有意差あり
ランダム	4877.52 ± 300.31	82.8%	期限順・RLと同等
強化学習	4790.38 ± 299.92	74.6%	期限順・ランダムと同等

考察

仮説「強化学習が最も優れたスケジューラーになる」は実証されなかった。統計的には期限順・ランダム・強化学習の3手法の間に有意差は見られず（期限順 vs RL: p=0.8353、ランダム vs RL: p=0.1498）、ほぼ同等のパフォーマンスとなった。

重要度順の高スコアと低完了率について、重要度順は最高スコア（5037.80）を記録したが、完了率は59.0%と最低だった。HIGHタスクは長時間かつ疲労係数が大きいため、重要なタスクから取り組むと集中力がすぐに低下し、その後のタスクの消化ペースが落ちる。スコア指標では有利だが、タスクをこなせた「量」では劣る結果となった。

強化学習が期待通りに機能しなかった原因として、以下が考えられる。

1. 状態空間の粗い離散化：学習効率のために各次元の区間数を絞ったが、集中力・疲労などの細かな差異を状態として区別できなかった可能性がある
2. 報酬関数の複雑さ：複数の報酬要素を加算した結果、正しい行動に対して報酬が一貫して与えられず、学習の方向性がブレた可能性がある
3. 学習量の不足：20,000エピソードでも、8次元の状態空間を十分に探索しきれていない可能性がある

今後の改善方向として、DQN（Deep Q-Network）など関数近似を用いた手法の導入、報酬関数のシンプル化、より多くのエピソードでの学習（50,000以上）、実際のユーザーの行動データを活用したQテーブルの更新が挙げられる。

感想

強化学習を実際に実装して動かすのは初めての経験だった。Q-learningのアルゴリズム自体は理解していたものの、状態空間の設計や報酬関数のチューニングが実際にはとても難しく、「どの状態表現がエージェントにとって意味のある情報か」という問いに対して何度も設計を見直した。

また、実験結果が仮説を支持しなかったことは正直悔しかったが、「なぜ強化学習が機能しなかったのか」を考察する過程で、機械学習モデルの性能はアルゴリズム以上に状態設計・報酬設計・学習量に大きく依存することを実感した。理論を理解するだけでなく、実際に試して結果を観察することの重要さを感じた経験だった。