CSE4200-01

추천 시스템

PPO-based RS

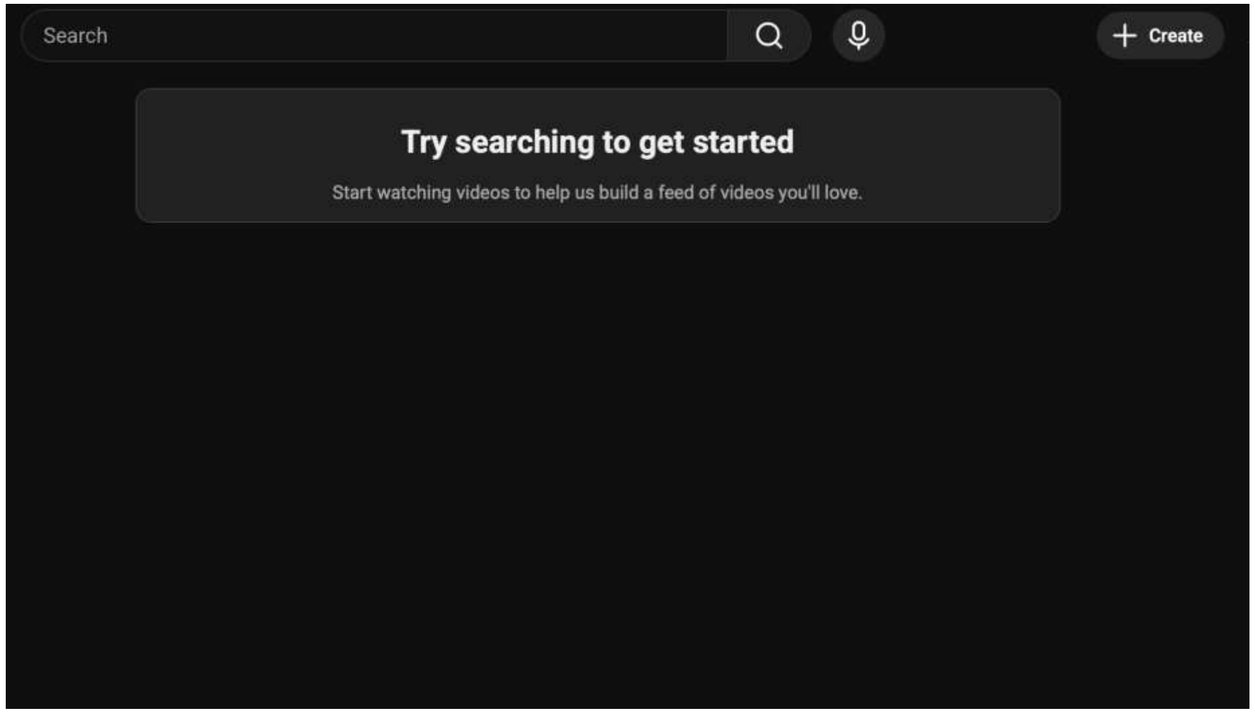
김대원 한유승 유승훈 강윤지 이채원

목차

01 Background & Motivation

User-Cold Start Problem

서비스를 처음 사용하는 사람에게 무엇을 추천해줘야 하는가?



History Data의 부재

사용자와 아이템 간의 상호작용 정보 없음

그렇기에 개인에 맞춘 추천이 어려움

사용자의 선호도를 상호작용하며 유추하여

최대한 잘 추천해줄 수 있다면?

01 Background & Motivation

순차적 추천 시스템

유저와 시스템의 상호작용을 정적인 집합이 아닌 순서가 있는 시퀀스로 모델링 하는 추천 시스템

Markov Chain 내지는 GRU4REC (2016) 과 같이 RNN 구조를 통해 주로 모델링

01 Background & Motivation

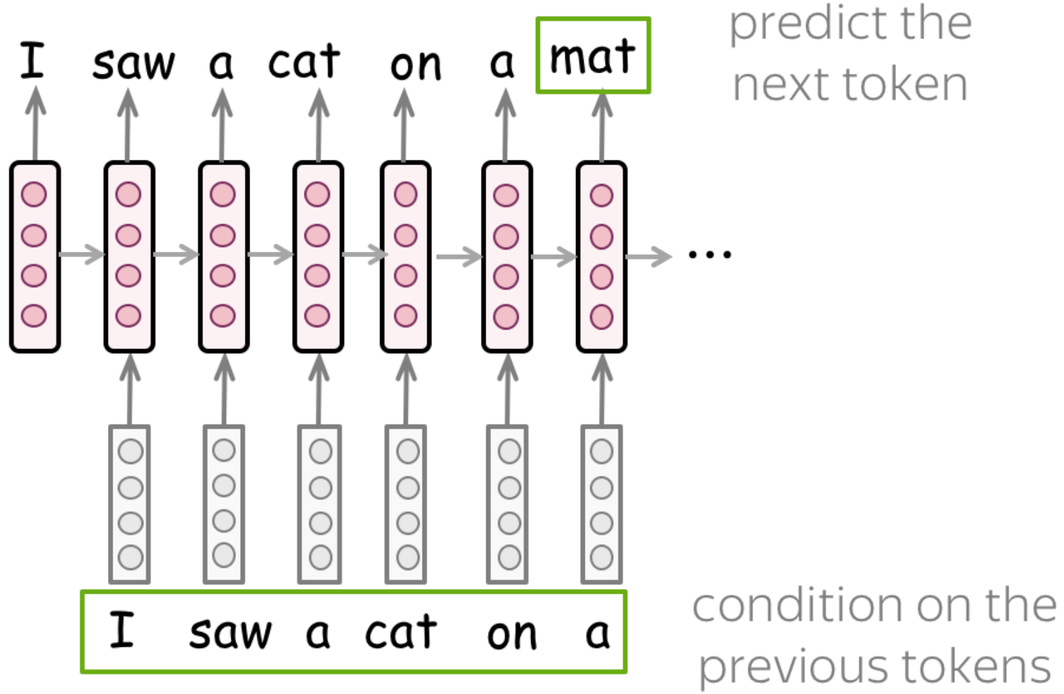
순차적 추천이 User Cold-Start Problem 을 완화 시키는 이유

Item-to-Item Transition

User ID 의존성에서 벗어날 수 있음

어떤 아이템이 특정 아이템 다음으로
소비되는지 전이 패턴을 학습

이력이 전혀 없는 소비자도 단기적 의도를
빠르게 파악할 수 있음



01 Background & Motivation

순차적 추천이 User Cold-Start Problem 을 완화 시키는 이유

Item-to-Item Transition

User ID 의존성에서 벗어날 수 있음

어떤 아이템이 특정 아이템 다음으로
소비되는지 전이 패턴을 학습

이력이 전혀 없는 소비자도 단기적 의도를
빠르게 파악할 수 있음

김영한의 자바 입문
코드로 시작하는 자바 첫걸음
처음 배우는 자바 기초

김영한의 실전 자바 - 기본편
객체지향 프로그래밍
코드로 익히는 OOP

김영한의 실전 자바 - 중급 1편
실무를 위한 다양한 중급 기능

01 Background & Motivation

순차적 추천 시스템의 한계 : 장기적 최적화 불가능한 Greedy Algorithm

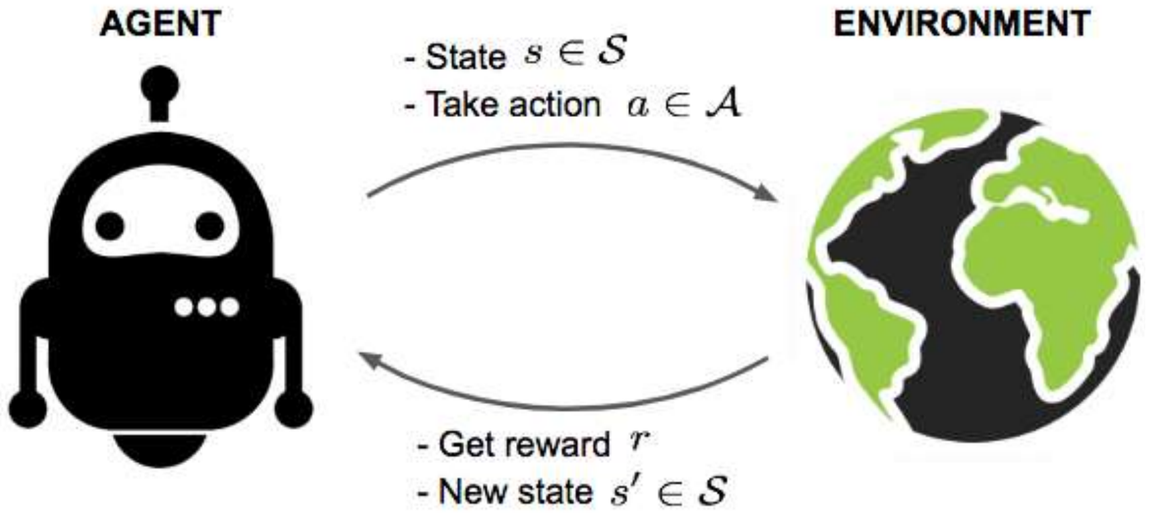
자극적인 쇼츠 영상을 클릭하면 다음에도 자극적인 영상 추천해주지만

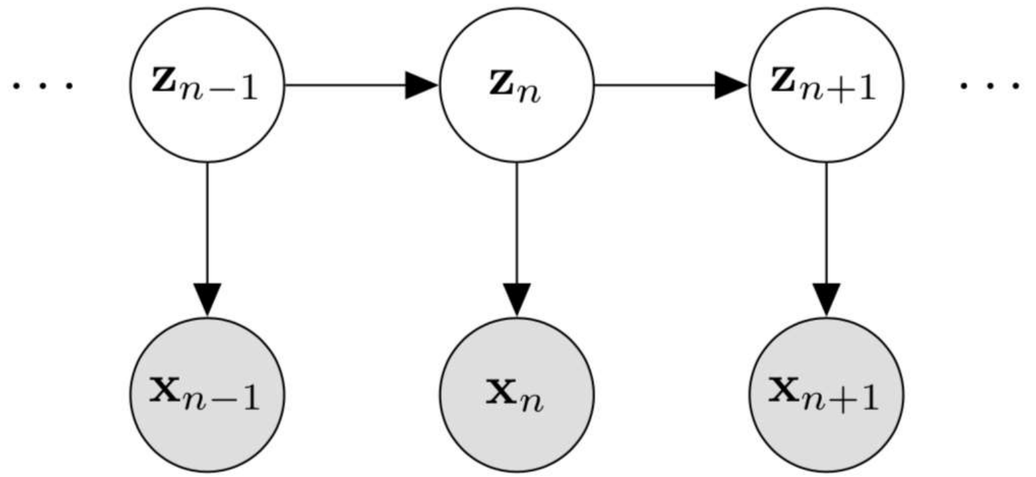
사용자가 피로도를 느끼거나 비슷한 컨텐츠만 추천하여 사용자 이탈로 이어질 수 있음

01 Background & Motivation

장기적 보상 극대화 : 강화학습을 활용한 최적화

사용자 (Environment) 와 추천 시스템 (Agent)의 상호작용을 통해, 사용자의 효용을 보상으로 설정하여 장기적 총 효용을 극대화 시킬 수 있도록 강화학습을 활용하여 모델링





사용자의 실제 선호도는 직접 알 수 없기에, 사용자와 아이템 간의 상호작용을 통해 이를 간접적으로 유추해야 함

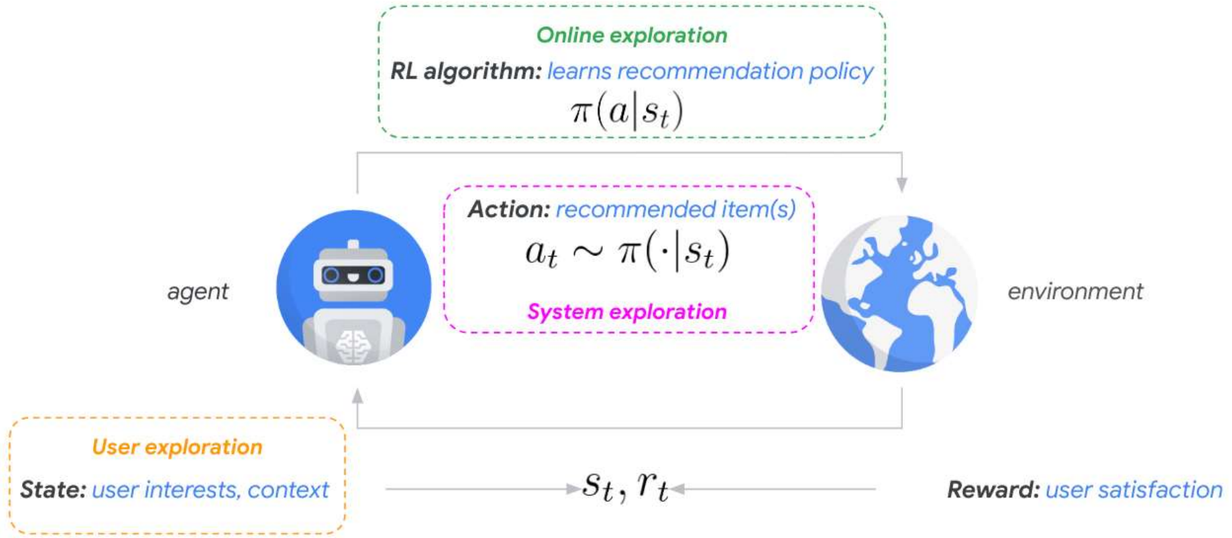
(Partially Observable Markov Decision Process 로 모델링)

02 Project Goal

강화학습에 기반한 장기적 총효용 극대화 SRS 구축

Sequential Recommender System + Reinforcement Learning

기존의 순차적 시스템에 강화학습을 접목 시켜 Click-Through-Rate 뿐 아니라 장기적 사용자 만족도를 극대화 시킨다



02 Problem Definition

Slate Recommendation with Reinforcement Learning

What to Recommend : Slate

해당 프로젝트에서는 슬레이트(Slate) 추천

Slate 란?

쉽게 말해 K개의 추천 목록을 뜻함

기존 강화학습에서는 하나의 item만 추천

이를 모델링 할 필요 O

[Slate for YouTube Shorts]

02 Problem Definition

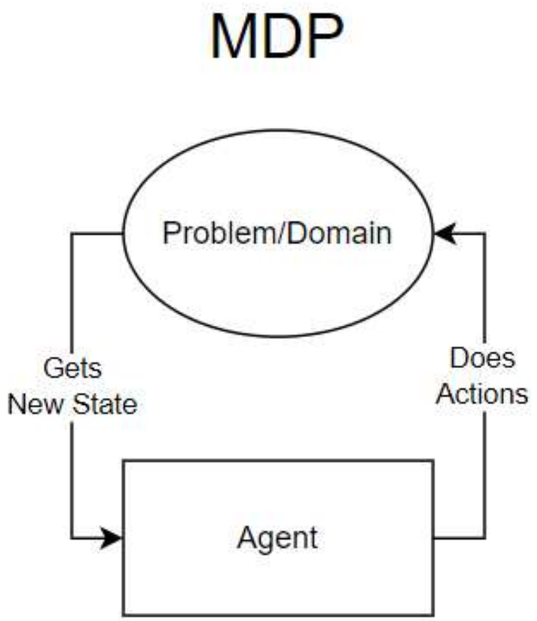
Slate Recommendation with Reinforcement Learning

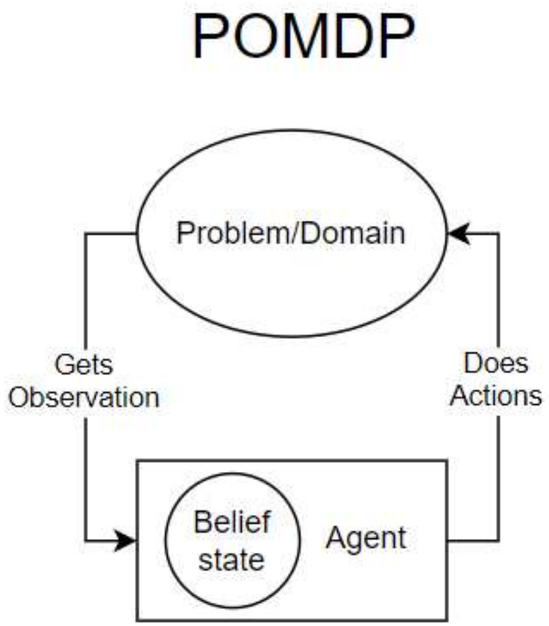
Markov Decision Process

에이전트가 환경의 진짜 상태를 완벽히 관측 가능하여 Markov Property 만족할 때의 장기적 효용 극대화 의사결정 과정

Partially Observable MDP

상태가 부분적으로만 관측되어 Belief State 추적하여 숨겨진 상태 추론하는 MDP 방식





사용자 (Environment) 의 Profile 을 알 수 없기 때문에 해당 프로젝트에서는 POMDP 로 추천 시스템을 모델링하여 처리

GRU, DeepSets 를 Backbone 모델로 하여 PPO 등의 알고리즘으로 Agent 를 학습시킨다

02 Problem Definition

Slate Recommendation with Reinforcement Learning

Simulator for Online Learning

Online User Study 는 지나치게 비용이 큼

강화학습은 환경과의 상호작용이 필수적

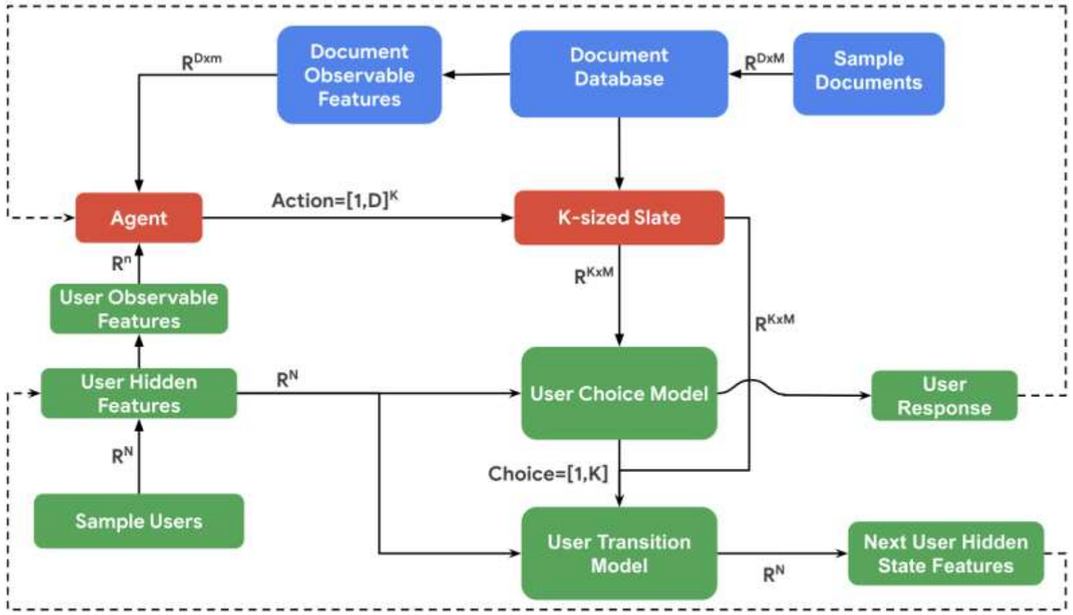
이를 위해 2019년 Google 에서 개발한

RecSim 을 사용

User Interest Evolution Model

사용자는 최근 상호작용한 아이템과 유사한

아이템들을 선호하도록



N - number of features that describe the user's hidden state
n - number of features that describe user's observed state
M - number of features describing document hidden state
m - number of features describing document observed state
D - total number of documents in the corpus
K - size of slate

Figure 1: Data Flow through components of RecSim.

03 Methodology

POMDP Modeling for Slate Recommendation

Deep Sets

집합을 임베딩 하기 위해 만들어진 아키텍쳐

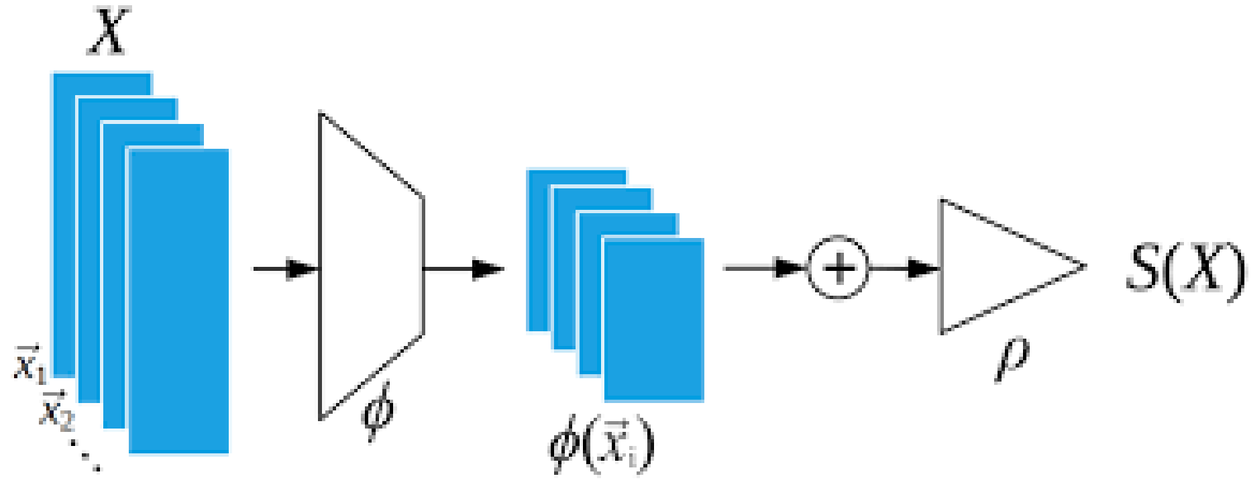
이전에 사용자에게 추천한 Slate 를 Set으로
간주하여 이를 임베딩

LightGCN

GCN 변형 GNN 기반 모델

초기 로그 데이터로 Item-Embedding 학습

[Deep Sets Architecture Diagram]

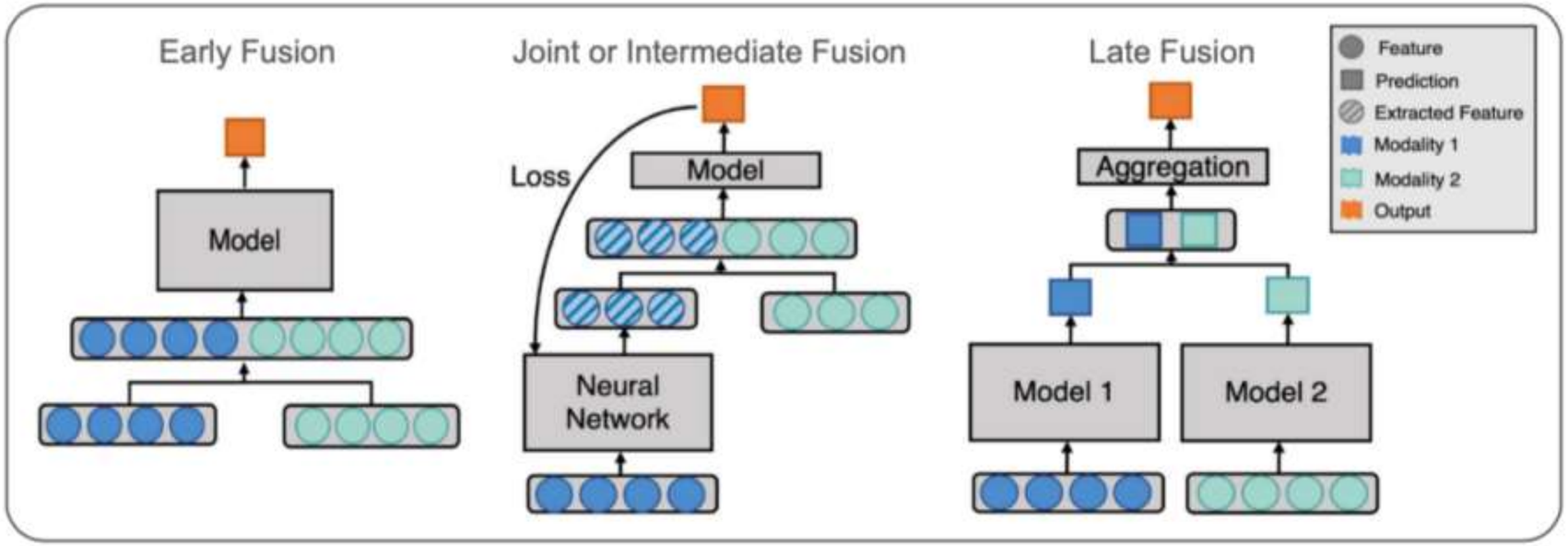


03 Methodology

POMDP Modeling for Slate Recommendation

각 시기마다 Agent는 직전 시기에 추천한 Slate 와 User가 선택한 Item 을 Late Fusion 형태로 처리함.

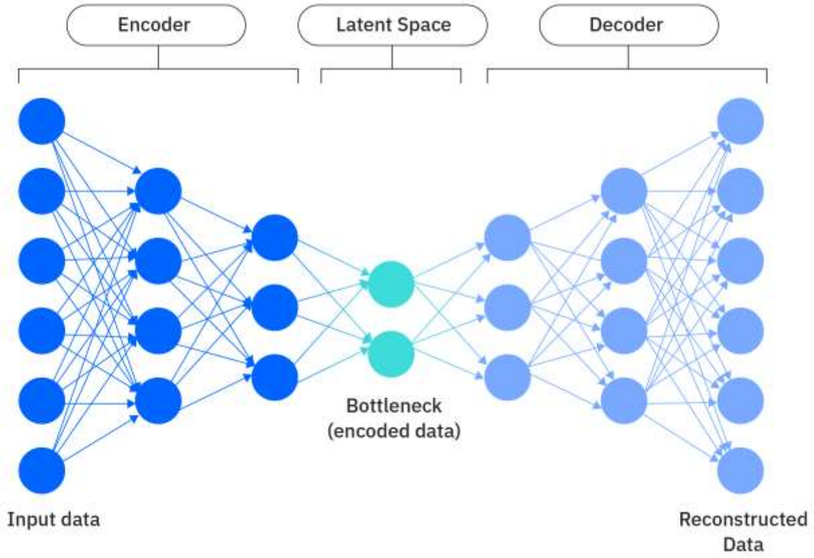
이를 RNN의 입력으로 사용하여 Agent 는 Hidden State 를 업데이트함 (Belief State Update 와 동일한 기능)

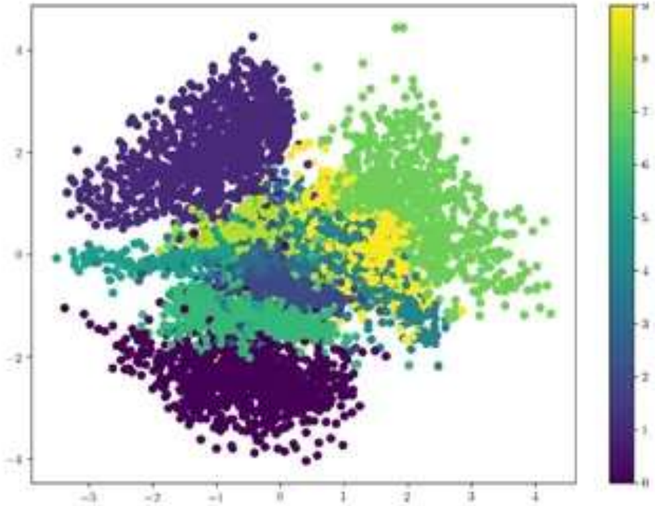


03 Methodology

Slate Generation with Classifier, Not Proto-Action

초기 제안 : Action Selection을 Item-Embedding 과 동일 선상에서 KNN 알고리즘으로 Slate Generation





이와 유사한 방식인 Wolpertinger Architecture 로 Diversity Penalty 를 넣고 처리하였으나 Item Embedding 의 Latent Manifold 와 Mismatch 가 일어나 제대로 된 Slate 를 만들 수 없었음

03 Methodology

Knowledge Distillation Step : GRU4REC as a Teacher

Knowledge Distillation

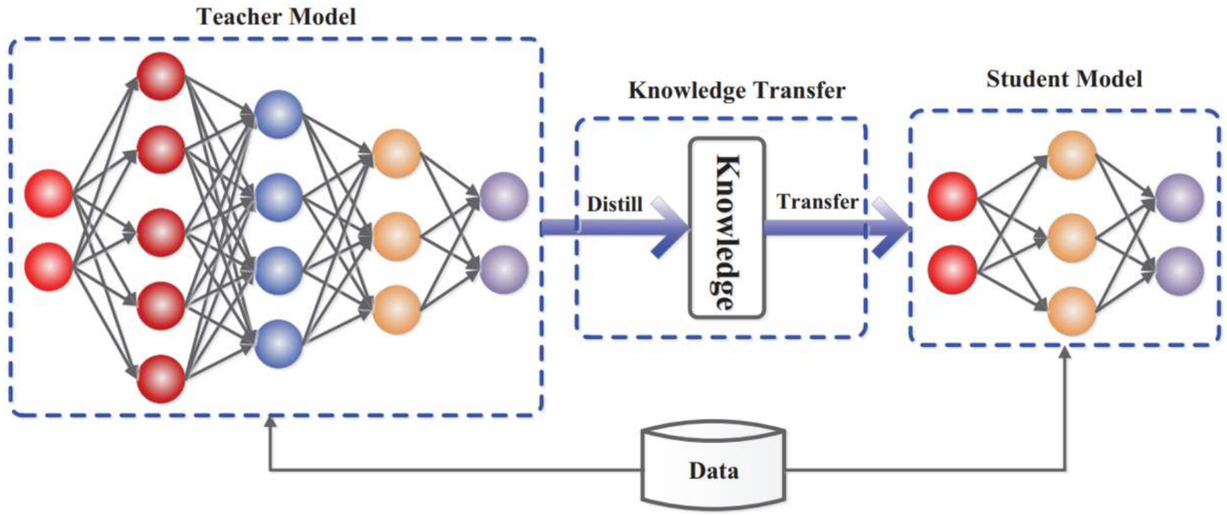
GRU4REC 을 교사 모델로 Agent 의 Slate Sampler 학습 시킴 (KL-Divergence)

Why KD is required ?

Exploration – Exploitation Trade-off

생각하면 Exploration 를 많이 줄이게 되는 결과를 초래하지만 학습의 안정성을 위함

[Knowledge Distillation]



03 Methodology

Proximal Policy Optimization for Stable Learning

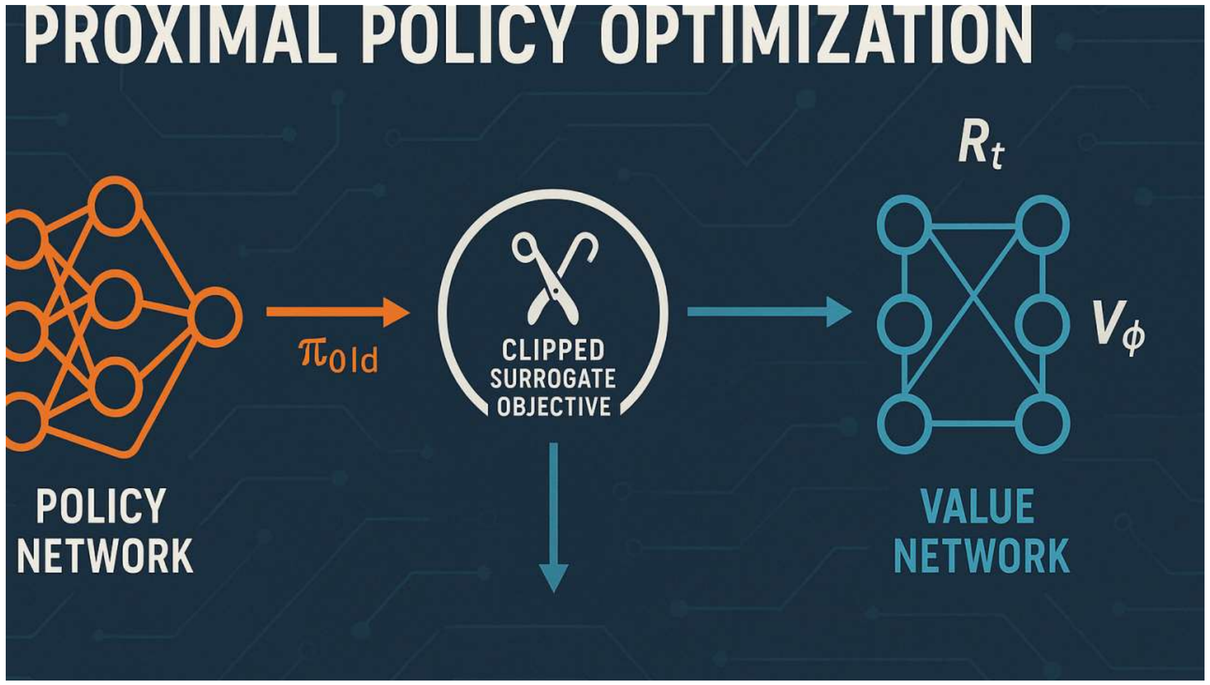
Clipped Surrogate Objective

학습의 안정성을 위해 이전 분포와 큰 차이
없도록 학습함

Actor-Critic with V(s)

Q(s, a) 의 경우 action이 Slate 이므로
계산이 어려움. Value Network로 해결.

Slate-Q Decomposition 으로 처리하더라도
User-Choice Model 등 CTR 예측기 필요



04 Data Collection

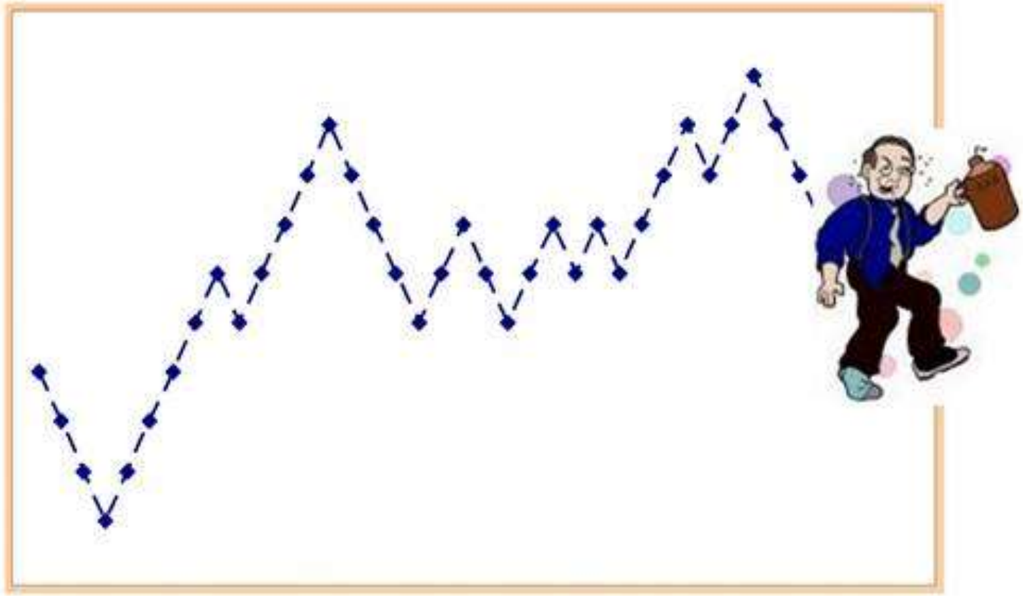
Independent and Identically Distributed Samples

Random Walk

추천 시스템이 없는 경우 사용자는 현실에서
랜덤한 아이템들을 마주하게 되고 소비함

특정한 추천 정책을 기반으로 데이터를 생성할 경우
각 데이터 샘플 간 상관 관계가 발생하여 bias 발생

이때, 무조건 Retention 하도록 강제한 환경에서
데이터 생성, 각 사용자 별 timestep 길이 일정



04 Data Format

Rating Matrix for Training Item Embedding (e.g. LightGCN)

	i_1	i_2	i_3	⋯	i_N
u_1	0	1	0	…	2
u_2	2	0	4	…	5
u_3	3	0	0	…	0
⋮	⋮	⋮	⋮	…	⋮
u_M	0	0	2	…	3

$\mathbb{R}^{M \times N}$

(a) Explicit feedback

	i_1	i_2	i_3	⋯	i_N
u_1	0	1	0	…	1
u_2	1	0	1	…	1
u_3	1	0	0	…	0
⋮	⋮	⋮	⋮	…	⋮
u_M	0	0	1	…	1

$\mathbb{R}^{M \times N}$

(b) Implicit feedback

04 Data Format

User-Trajectory for Training Sequential Model

05 Benchmark (500 x 500)

Static Dataset for LightGCN : 30 steps per user

Life-time Return

한 사용자가 시스템과 상호작용하는 전체 세션동안 **누적된 보상의 현재 가치의 합.**
단기 최적화 극복 문제 평가 지표로 채택

CTR

추천 시스템이 사용자에게 추천한 아이템 목록 중에서 아이템을 선택할 확률

Coverage

전체 중 추천된 아이템의 비율
Agent의 **탐색 품질** 및 **추천 편향성** 관찰 지표로 채택

Episode Length

각 에피소드 별 timestep count
유저가 얼마나 오랫동안 서비스를 이용하는지 측정하는 지표
(Retention Proxy)

05 Benchmark (500 x 500)

Baseline Algorithm 1 : Content-based Filtering

Content-based Filtering

기존의 CBF 방식은 사용자의 Embedding을 상호작용한 Item-profile 들을 보고 Incremental Mean 내지는 Arithmetic Mean 을 이용해서 학습하게 됨

User Cold-Start Problem Mitigation 을 위해 학습 데이터셋에 있었던 모든 User-Profile의 평균 시작 이후 Incremental Mean 형태로 업데이트 : $\text{use}_t \coloneqq \alpha\, user_{t-1} + (1-\alpha) item_t$ (실험 Setting : 0.1)

Drift Scale	Average Reward	Episode Length	Coverage	CTR
0.1 (Stable)	157.55	70.20	0.98	0.57
0.5 (High)	155.57	70.31	0.98	0.56
1.0 (Drastic)	151.24	78.47	0.98	0.49

05 Benchmark (500 x 500)

Baseline Algorithm 2 : GRU4REC

Next-Item Prediction-based Top-K

Random-Walk Generated Sequence 를 통해 Next-Item Prediction 형태로 학습

이후, Top-K Slate Recommendation 으로 맞춤형 추천 수행

Item Token Embedding 으로 Implicit User-Item Matrix 로 학습한LightGCN Embedding 사용

Top-1 Loss 를 통해 학습 (논문 기본 설정)

Drift Scale	Average Reward	Episode Length	Coverage	CTR
0.1 (Stable)	176.09	64.80	1.00	0.69
0.5 (High)	173.13	66.42	1.00	0.66
1.0 (Drastic)	174.12	70.58	1.0	0.62

05 Benchmark (500 x 500)

Proposed Algorithm : PPO-based Recommendation

Policy Network

Classifier 와 같이 logit 을 계산함. 이후 Top-K slate generation 하여 추천

심각한 수준의 Mode Collapse 발생

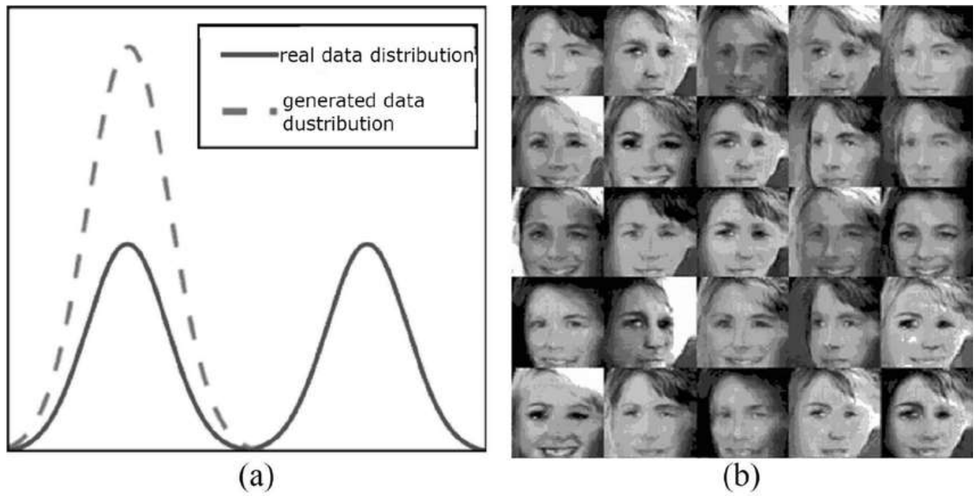
대중적으로 가장 인기가 많은 제품들만 반복적으로 추천

(Coverage 0.11, 0.10, 0.10)

초기 생성형 모델인 GAN 에서도 이와 동일한 문제 발생

GAN 의 경우, 생성자가 판별자를 속이는 형태로 학습

실제 데이터 분포와 달리, 자신이 만들기 쉬운 이미지만 만드는 Mode Collapse 발생



05 Benchmark (500 x 500)

Proposed Algorithm : PPO-based Recommendation

Soft label KL-Divergence Loss Penalty

Over-fitting 을 방지하고 Coverage 를 높이기 위해, 사용자가 선택한 item 과 유사한 아이템 임베딩 활용해서 유사한 Item 을 더 많이 보도록 추가 (Weak Supervised Guidance)

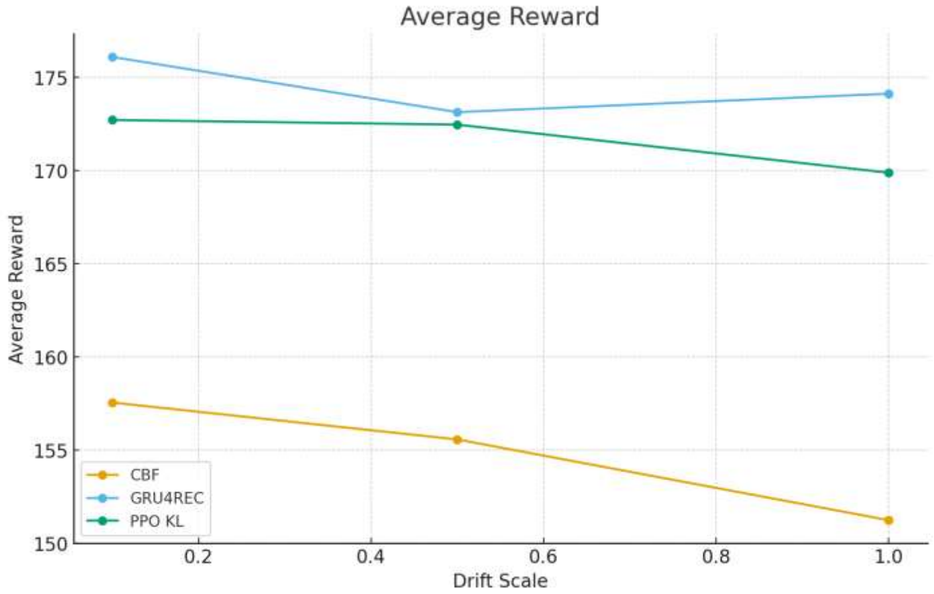
$$D_{KL}(p(x)||q(x)) = \int_{-\infty}^{\infty} p(x) ln \frac{p(x)}{q(x)} dx$$

Drift Scale	Average Reward	Episode Length	Coverage	CTR
0.1 (Stable)	172.71	73.76	0.13	0.59
0.5 (High)	172.46	76.45	0.13	0.57
1.0 (Drastic)	169.88	85.23	0.13	0.50

Coverage 를 소량 증가, 가장 긴 Episode Length (Mode Collapse 여전히 존재)

05 Benchmark (500 x 500)

Over-All Results : GRU4REC is the Winner



[Average Cumulative Reward]

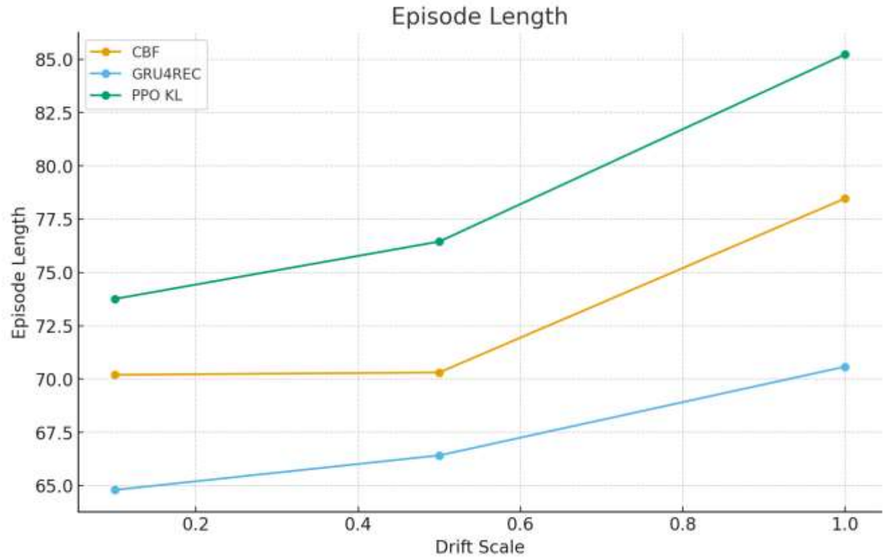
Drift-Robustness

GRU4REC 은 Sequence Modeling 을 통해 사람들의 소비 패턴을 학습함.

PPO-based Recommender 는 GRU4REC 대비 성능 하락 폭이 작지만 CBF는 상당히 낮은 Average Return

05 Benchmark (500 x 500)

Over-All Results : GRU4REC is the Winner



[Average Episode Length]

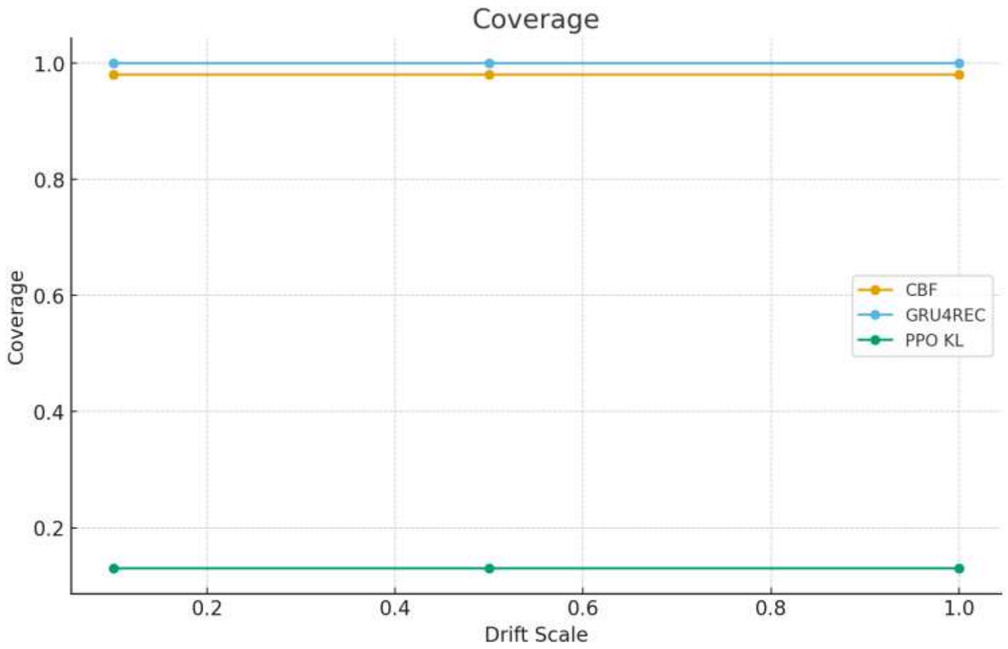
User Retention (사용자 유지)

Background & Motivation 에서 말했 듯, 순간적인 만족도는 극대화 할 수 있으나, 빠른 사용자 이탈로 이어질 수 있다는 문제점 존재

PPO 는 POMDP 방식으로 모델링 되므로 Episode Length 가장 길은 특성 보임

05 Benchmark (500 x 500)

Over-All Results : GRU4REC is the Winner



[Coverage]

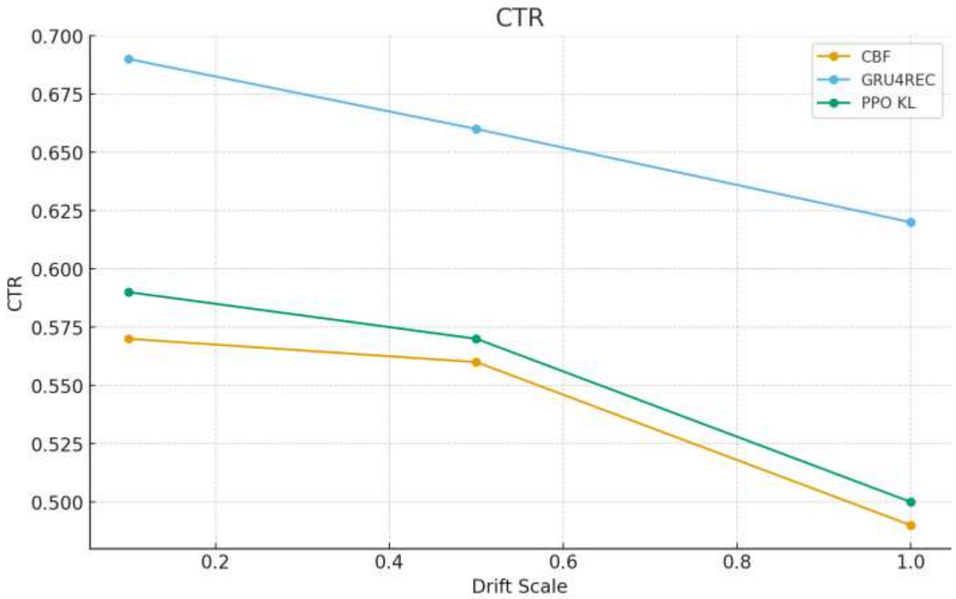
Mode-Collapse of PPO

PPO-based Algorithm 의 경우 가장 대중적인
소수의 아이템을 집중적으로 추천함

강화학습의 경우 Reward 가 Sparse 한 경우
Mode Collapse 가 쉽게 발생함

05 Benchmark (500 x 500)

Over-All Results : GRU4REC is the Winner



[Click-Through Rate]

Click-Through Rate

사용자가 실제로 선택하는 아이템들을 기준으로
학습하는 것이기에 CTR 가 제일 높음

하지만 Average Episode Length 가 가장 낮음
(Background & Motivation 에서 말한 Greedy 특성)

05 Conclusion

Limitation & Future Work : Proto-Action with VAE

Proto-Action for Latent Manipulation

Latent Vector Sampling 을 위해 VAE 학습시키기.

이를 통해 PPO, SAC 등의 알고리즘으로 Latent Manipulation 시키기. (Imitation Learning 분야)

Better Backbone Model for Embedding

Deep Sets → Set Transformer, Classifier → SetVAE, GRU → Transformer

06 Each Member's Role

김대원

- 연구 주도 및 실험 설계, 결과 해석, 보고서 작성
- Colab Pro+ A100 GPU 활용한 학습 가속화

한유승

- RecSim TF 를 최신 라이브러리에 맞춰 Refactoring
- API 형태로 즉각적으로 데이터 추출 가능하게 구현

유승훈

- 프로젝트 파일 계층 구조 조직화 및 **실험 재현 코드 작성**
- RecSim TF 를 최신 라이브러리에 맞춰 Refactoring

강윤지

- LightGCN 구현 및 임베딩 학습
- CBF Hybrid 추천 정책 구현

이채원

- RecSim ↔ PPO Agent 학습 파이프라인 구현
- Knowledge Distillation 기반 Offline RL 학습

References

[1] RecSim: A Configurable Recommender Systems Simulation Platform

[2] Proximal Policy Optimization Algorithms

[3] Deep Sets

[4] Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

[5] Session-based Recommendations with Recurrent Neural Networks

[6] Deep Reinforcement Learning in Large Discrete Action Spaces

07 질문과 답변

감사합니다

THANK YOU!