Integrando avaliações contrafactuais aos frameworks tradicionais de recomendação interativa*

Yan Andrade[1] **, Nícollas Silva**[2] **, Leonardo Rocha**[1]

[1] DCOMP/UFSJ - São João del-Rei, MG , Brasil
[2] DCC/UFMG - Belo Horizonte, MG , Brasil

yrandrade123@aluno.ufsj.edu.br, ncsilvaa@dcc.ufmg.br, lcrocha@ufsj.edu.br

Abstract. *Online recommendation task has been recognized as a Multi-Armed Bandit (MAB) problem. Despite the recent advances, there is still a lack of consensus on the best practices to evaluate such bandit solutions. Recently, we observed two complementary frameworks that allow us to evaluate bandit solutions more accurately: iRec and OBP. The first has a complete set of datasets, metrics and MAB models implemented, allowing only offline evaluations of these solutions. However, the second is limited to a few bandit solutions with more current metrics and methodologies, such as counterfactuals. In this work, we propose and evaluate an integration between these two frameworks, demonstrating the potential and richness of analyzes that can be carried out from this combination.*

Resumo. *A tarefa de recomendação online vem sendo reconhecida como um problema de Multi-Armed Bandit (MAB). Apesar dos avanços recentes, ainda há falta de consenso sobre as melhores práticas para avaliar essas soluções. Recentemente, observamos dois frameworks complementares que nos permitem avaliar soluções bandit com mais precisão: iRec e OBP. A primeira possui um conjunto completo de coleções de dados, métricas e modelos MAB implementados, permitindo apenas avaliações offline. Já o segundo se limita a algumas soluções bandit, porém com métricas e metodologias mais atuais, como os contrafactuais. Neste trabalho, propomos e avaliamos uma integração entre esses dois frameworks, demonstrando o potencial e a riqueza de análises que podem ser realizadas a partir dessa combinação.*

1. Introdução

Atualmente, diversas aplicações Web têm investido em Sistemas de Recomendação (SsR) para orientar toda a experiência dos usuário desde suas primeiras interações como um modelo de decisão sequencial [Zhou et al. 2020]. Nesse caso, na interação de cada usuário, o sistema deve recomendar um ou mais itens, receber o feedback do usuário e atualizar seu conhecimento para a próxima recomendação [Wu et al. 2018]. A ideia é aprender a cada interação para aumentar o conhecimento do sistema e maximizar a satisfação do usuário no longo prazo. Os trabalhos atuais abordaram esse desafio como um problema do Multi-Armed Bandit (MAB), onde os itens são modelados como *arms* a serem selecionadas, e a experiência do usuário é representada pelo *reward* acumulado [Shams et al. 2021].

*Esse trabalho foi parcialmente financiado por AWS, CNPq, CAPES, FINEP e Fapemig

Embora tenham havido avanços recentes, há uma completa falta de consenso sobre as melhores práticas de avaliação de um sistema de recomendação interativo. Tradicionalmente, as avaliações de novos algoritmos são realizadas offline utilizando um conjunto de dados pré-selecionado relacionado à recomendações de itens e suas respectivas avaliações feitas pelos usuários. Esse conjunto é dividido em treinamento, utilizado para treinar os recomendadores, e teste utilizado na avaliação dos mesmos. O desempenho dos algoritmos são medidos por meio de métricas como precisão e revocação. Para este cenário destacamos o *iRec* [Silva et al. 2022], um framework para avaliar sistemas de recomendação interativos fornecendo uma comparação justa entre distintos SsR com diversas metodologias de avaliação amplamente testadas e utilizadas na literatura. Contudo, avaliações offline sofrem com viés dos dados, uma vez todas as avaliações de usuários registradas foram coletadas com base em itens recomendados por uma política previamente implantada, resultando em uma reflexão imprecisa das reais preferências dos usuários [Yang et al. 2022].

Para evitar esse viés, SsR também podem ser avaliados por metodologias online, tais como testes A/B [Liu et al. 2022]. Todavia, essas metodologias demandam um maior tempo e esforço para obter resultados. Para contornar essas limitações, pesquisadores têm recorrido a abordagens de avaliações contrafactuais [Saito et al. 2020]. Nesse caso, a partir de dados pré-existentes como na avaliação offline, para os quais temos conhecimento da política de recomendação utilizada nas recomendações, estima-se como seria o desempenho de uma nova estratégia caso fosse utilizada em substituição à política de origem (estimadores). Assim, a avaliação contrafactual simula uma situação hipotética em que o usuário recebeu uma recomendação diferente e compara a probabilidade de tomar a ação desejada nessas duas situações, se assemelhando a um teste A/B offline [Saito et al. 2020]. Nesse cenário, temos o Open Bandit Pipeline (OBP), um framework que fornece um procedimento experimental completo e padronizado para avaliações utilizando a metodologia contrafactual. Uma vez que a maioria de coleções de dados públicas não apresenta qual a política de recomendação utilizada, o OBP incluí também módulos para a criação de dados sintéticos.

É evidente que, para a realização de avaliações abrangentes de SsR interativos, é necessário levar em consideração tanto as metodologias tradicionais, fornecidas pelo *iRec*, quanto as contrafactuais, oferecidas pelo OBP, suportando diferentes tipos de análise e permitindo a comparação precisa e objetiva de resultados. Assim, nesse trabalho propomos uma integração entre o *iRec* [Silva et al. 2022] e o OBP [Saito et al. 2020], resultando em uma ferramenta que realiza tanto avaliações offline, com as principais métricas, quanto avaliações online com diferentes estimadores. Avaliamos nossa proposta por meio de uma experimentação que considerou quatro coleções (três tradicionais e uma sintética), cinco SsR distintos, três métricas de avaliação tradicionais e três estimadores contrafactuais. Por meio desse experimento, foi possível avaliar não apenas o desempenho dos algoritmos em si, mas também o impacto de fatores externos, como o contexto do usuário e a interação do recomendador com o usuário ao longo do tempo.

Todas as implementações e execuções dos experimentos foram realizadas pelo aluno Yan Andrade, sob a orientação do professor Leonardo Rocha. A concepção do projeto e as análises de resultados foram feitas em conjunto, aluno e professor, com a colaboração do doutorando do Programa de Pós-Graduação do Nícollas Silva.

2. Referencial Teórico

2.1. Multi-Armed Bandits

Na literatura, o problema de multi-armed bandits [Auer et al. 2002] é definido como um modelo de decisão sequencial, no qual continuamente escolhe uma ação a pertence ao conjunto de ações A, conhecidos como braços (*arms*). Essa seleção é guiada pela política de cada modelo e pela função de valor relacionada à importância de cada braço. Ao selecionar uma ação $a \in A$ na interação t resulta em uma certa recompensa $R_t(a_t)$. Tendo como objetivo principal decidir em uma sequência de ações que maximize as recompensas após T interações: $\sum_{t=1}^{T} R_t(a_t)$.

Se o sistema tiver conhecimento suficiente sobre o domínio, a melhor opção é selecionar a ação que proporciona a máxima recompensa possível a todo tempo [Sanz-Cruzado et al. 2019]. Contudo, esse conhecimento é incerto e muitas vezes desconhecido. Por isso, o modelo MAB tem sempre que decidir entre duas opções. Uma primeira opção, mais conservadora, é selecionar os *arms* com as maiores recompensas do passado – uma abordagem de *exploitation*. Em contrapartida, outra opção é investir em diferentes *arms* a fim de obter mais informações sobre o domínio e tomar decisões futuras ainda melhores – uma abordagem de *exploration*. Tais opções caracterizam o dilema de exploitation-exploration (i.e. exp-exp) e exigem que o modelo seja capaz de explorar o máximo conhecimento disponível enquanto também explora o espaço de solução para adquirir ainda mais conhecimento sobre o domínio [Sanz-Cruzado et al. 2019].

Nesse sentido, a qualidade de uma solução MAB está relacionada à forma como ele trata esse dilema de exploration e exploitation. Diversas estratégias foram desenvolvidas [Silva et al. 2022]. **ϵ-Greedy** é um modelo clássico de bandit que explora aleatoriamente outros arms com probabilidade ϵ. **UCB** calcula intervalos de confiança para cada item e a cada interação tenta reduzir os limites de confiança. **Thompson Sampling (TS)** segue uma distribuição Gaussiana de itens e usuários para predizer baseado em amostras. **Linear TS** é uma adaptação linear do original TS para medir dimensões latentes usando Probabilist Matrix Factorization (PMF). **Linear ϵ-Greedy** realiza *exploitation* linear de fatores latentes definidos pelo PMF do clássico ϵ-Greedy. **Linear UCB** é uma adaptação do original LinUCB para mensurar as dimensões latentes pela formulação PMF. **GLM-UCB** é uma adaptação do Linear UCB com uma forma sigmoidal que realiza o exploration dependente do tempo. O **ICTR** é um modelo de regressão de tópicos que utiliza o TS e controla a dependência de itens por uma estratégia de aprendizado de partícula. **KNN Bandit** é uma variante do nearest-neighbours aplicada ao clássico algoritmo TS sem parâmetro. **NICF** é uma combinação entre redes neurais e filtros colaborativos que aplicam aprendizado nas preferencias dos usuários. **COFIBA** define um UCB para combiná-lo com agrupamento adaptativo de usuários e itens. **PTS** é uma formulação PMF para o original TS que aplica filtros de párticulas para guiar a exploração de itens ao longo do tempo. **Cluster-Bandit (CB)** é baseado em grupos para enfrentar o problema de cold-start. O *iRec* possui todas essas abordagens implementadas internamente.

2.2. Contrafactual

Embora existam diversas abordagens para resolver o problema, ainda não existe um consenso de qual a melhor forma para avaliação desses algoritmos. Métricas tradicionais têm servido como base para a comparação da eficiência de cada um, porém estudos

recentes mostraram como o viés de seleção e exposição [Pan et al. 2021], existente nas coleções, impactam o resultado da avaliação. Ambos ocorrem quando os dados utilizados para treinar o sistema não são representativos da população como um todo, resultando em uma reflexão imprecisa das preferências do usuário que leva a uma seleção enviesada dos itens que foram expostos. Isso acontece em conjuntos de dados offline porque todas as avaliações de usuários registradas foram coletadas com base em itens recomendados por uma política previamente implantada (ou seja, o modelo em produção no momento da interação do usuário). Os estimadores contrafactuais permitem o uso de dados de log existentes para estimar como alguma nova política de recomendação de destino (ou seja, uma nova abordagem) teria sido executada se tivesse sido usada em vez da política que registrou os dados. Ele permite uma avaliação sem política (Off-Policy Evaluation - OPE) semelhante a um teste A/B off-line imparcial. Existem diversos estimadores contrafactuais propostos na literatura [Liu et al. 2022]. **Direct method (DM)** usa um modelo para completar as recompensas ausentes, e utiliza as recompensas dos itens selecionados. **Inverse Propensity Score (IPS)** atribui pesos de importância a política atual com base nos valores originais do recomendador usado para gerar a coleção dataset. **Clipped IPS (CIPS)** é adaptação do original IPS, com limitação de grandes pesos com um parâmetro λ. **Self-Normalized IPS (SNIPS)** reescala o valor do original IPS, pela soma de todos pesos de importância. **Doubly Robust (DR)** combina o DM com IPS para reduzir a variância e trabalhar bem com pequenas amostras. O OBP possui todos esses estimadores implementados internamente.

3. Ferramentas de Avaliação para modelos MAB

3.1. iRec

O *iRec*[1] é um *framework* proposto para viabilizar o uso de modelos interativos, em especial aqueles baseados em modelos Multi-Armed Bandit, no domínio de recomendação. Conforme podemos observar na Figura 1, o *iRec* é composto de três componentes principais que abrangem todo o processo de experimentação: (1) a construção de um **Environment**; (2) a definição de **Recommendation Agent**; e (3) a definição de uma **Experimental Evaluation**. No componente *Environment* configuramos toda a estrutura dos dados a serem processados pelo framework. Nele, carregamos as bases de dados desejadas e definimos todos os módulos de preparação de dados a serem aplicados a elas, tais como as estratégias de pré-filtragem e divisão dos conjuntos de treino e teste. Atualmente, o *iRec* possui 17 datasets públicos relacionados a diversos cenarios de aplicação, tais como filmes, músicas, pontos de interesse, produtos e roupas. Por sua vez, no componente *Recommendation Agent* seleciona o modelo que será utilizado para definir o(s) melhor(es) item(ns) para cada usuário em cada iteração. Em outras palavras, é nesse componente que implementam-se os SsR que serão utilizados na recomendação. Por fim, o componente *Experimental Evaluation* é responsável por realizar a integração dos modelos propostos no *Recommendation Agent* sobre os dados especificados no primeiro componente - *Environment*. Primeiramente, definimos a interação entre esses dois componentes por meio de um módulo denominado *Evaluation Policy*. Nele, configuramos um ambiente no qual um item (ou um conjunto de itens) é recomendado em cada iteração do algoritmo (uma tentativa). Por um tempo pré-determinado (número de

[1]Disponível em: `https://github.com/irec-org`

tentativas), o *Recommendation Agent* realiza as recomendações dentro do *Environment*, recebendo uma recompensa positiva ou negativa e atualizando seu conhecimento. Todos os *logs* (i.e., registro das ações realizadas pelo *Recommendation Agent* e recompensas fornecidas pelo usuários, de acordo com os dados no conjunto de teste) são armazenados. Por fim, o *Experimental Evaluation* analisa esses logs, aplica as métricas de avaliação de recomendações e realiza os testes estatísticos necessários para realizar uma avaliação adequada do desempenho dos SsR implementados no *Recommendation Agent*.

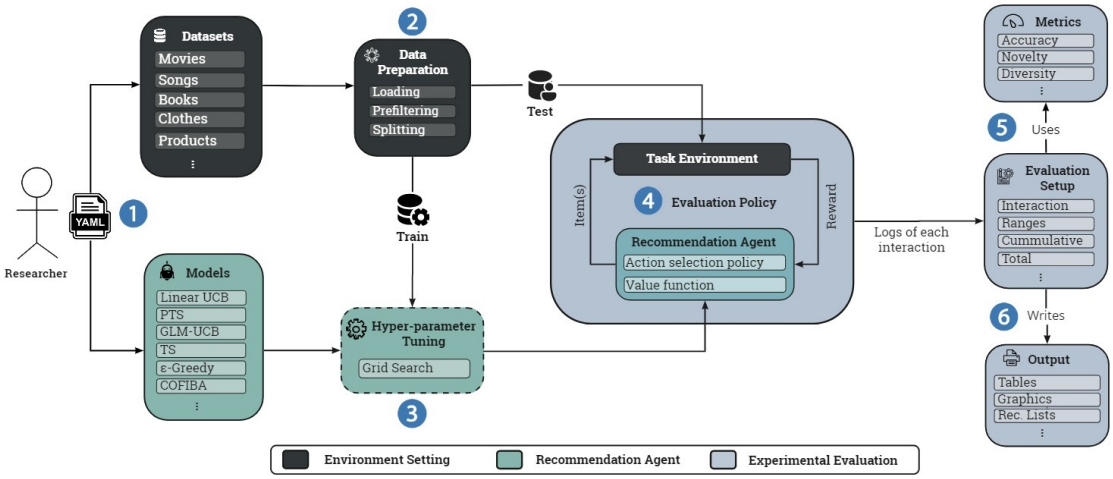

Figura 1. Uma visão geral da estrutura iRec [Silva et al. 2022].

3.2. Open Bandit Pipeline - OBP

O OBP [Saito et al. 2020] é uma biblioteca que inclui uma série de módulos para implementar o pré-processamento de coleções de dados e diversos métodos de recomendação. Diferente do *iRec*, essa biblioteca tenta lidar com o viés das bases de dados existentes para o cenário de recomendação. Atualmente, os *datasets* utilizados durante a fase de treinamento do modelos de recomendação são basicamente logs provenientes de um determinado recomendador, contendo informações sobre as interações entre usuários e itens. No entanto, sabemos que o processo de recomendação muda a forma com que os usuários interagem com o sistema, seja por clicks, avaliações e etc. Dessa forma, ao usar esses dados de interação dos usuários, estamos ignorando a natureza interventiva das recomendações. Como resultado, não estamos avaliando se os usuários clicariam ou comprariam mais devido às nossas novas recomendações, mas sim até que ponto as novas recomendações se ajustam aos dados registrados, e é justamente esse problema que o OBP tenta minimizar através de uma política de avaliação Contrafactual. Na Figura 2 é apresentada uma visão geral desta biblioteca, na qual podemos ver toda estrutura e seus principais módulos. Em suma, esta biblioteca pode ser dividida em quatro módulos principais: (i) módulo de dados, que fornece mecanismos para trabalhar nas coleções de dados, desde a etapa de carregamento dos dados, etapas de pré-processamento básicas, até métodos para gerar dados sintéticos, dentre outros; (ii) módulo de políticas que oferece interfaces para implementação de novos métodos *bandits* e novas políticas de avalição, além de já possuir diversos modelos e políticas de avaliações relevantes da literatura, tanto para o cenário online quanto off-line; (iii) módulo de simulação que fornece funções para para realizar simulação off-line de modelos *bandits*. Por meio dele é possível comparar e avaliar o desempenho de algoritmos MAB; e (iv) módulo de política

de avaliação que possui interfaces abstratas genéricas, ideais para implementações personalizadas, nas quais os usuários podem adicionar e novos estimadores, além de possuir estimadores clássicos e avançados já implementados, conforme vimos na Seção 2.2.

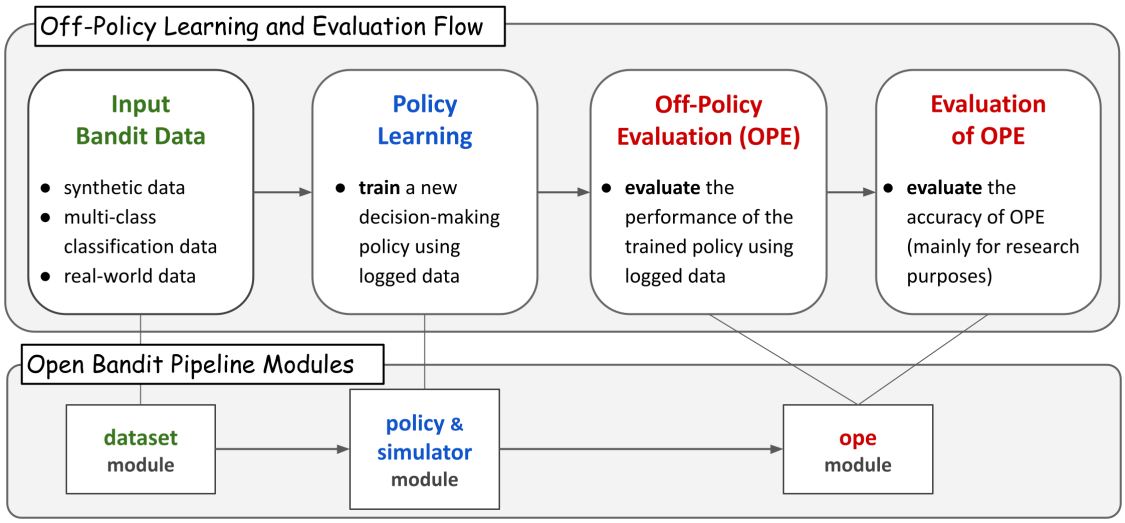

Figura 2. Visão geral da biblioteca Open Bandit Pipeline [Saito et al. 2020].

3.3. Ferramenta de integração

Embora o iRec e OBP já sejam bastante úteis individualmente, há um potencial inexplorado em combiná-los. O iRec oferece implementações dos principais recomendadores conhecidos na literatura, com módulos para simulação e avaliação offline usando métricas tradicionais. Por sua vez, o OBP possibilita um ambiente para a avaliação usando os principais estimadores contrafactuais por meio da geração de dados sintéticos com um recomendador base e o cálculo das probabilidades de cada item nas interações. É possível simular os dados sintéticos gerados pelo OBP nos recomendadores do iRec e avaliar o desempenho tanto online (usando o OBP) quanto offline (usando o iRec). Assim, propomos uma ferramenta capaz de integrar esse dois frameworks para realizar uma análise completa de recomendadores implementados como um modelo de Multi-Armed Bandit. Essa ferramenta fornece uma metodologia padronizada para a implementação de pré-processamento de dados, seleção dos métodos de aprendizado, e uma completa avaliação por meio das métricas tradicionais e estimadores contrafactuais (Off-Policy Evaluation - OPE). A unificação desses frameworks possui três módulos principais: (1) a construção do dados (OBP); (2) a simulação dos recomendadores (iRec); (3) a avaliação experimental (OBP e iRec).

O primeiro módulo, utiliza-se do OBP, e consiste na preparação dos dados para a execução dos experimentos. Isso inclui a criação, organização e parametrização dos dados sintéticos em um formato apropriado para análise. Para melhorar a flexibilidade e a capacidade de adaptação da ferramenta, foram adicionados três novos parâmetros à etapa de construção do dataset, além dos que o OBP já disponibiliza. O primeiro parâmetro permite indicar a quantidade de usuários distintos que serão simulados na análise, possibilitando a criação de datasets com diferentes tamanhos. Já o segundo parâmetro permite definir o vetor de contexto de cada usuário, que é uma informação adicional usada pelos modelos para realizar recomendações mais precisas. O último parâmetro

permite definir a frequência dos usuários. Com a adição desses parâmetros, é possível simular cenários mais complexos e realistas na avaliação de recomendadores em modelos MAB. A separação dos dados em treino e teste, é realizado por meio do iRec, com diferentes possibilidades. No segundo módulo, os dados preparados são utilizados para treinar e testar os modelos de recomendadores implementados pelo iRec. O framework é altamente customizável e oferece uma API intuitiva que facilita a integração com outros softwares. Ao utilizar o iRec neste segundo módulo, é possível avaliar diferentes modelos de recomendadores baseados em MAB. No terceiro módulo é feita a avaliação dos algoritmos de recomendação implementados, a partir dos logs gerados no módulo anterior. Para isso, são usadas técnicas de avaliação de desempenho MAB, que dividem-se em dois grupos: (1) as métricas tradicionais, como precision e recall, executadas por meio do iRec; e (2) os estimadores contrafactuais, como DM, IPS e DR, realizados pelo OBP.

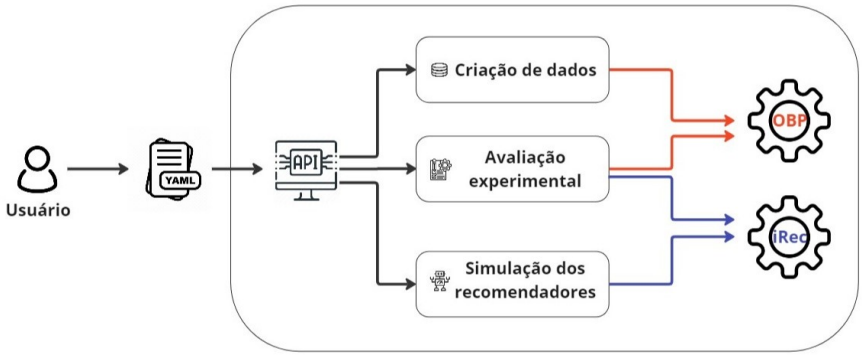

Figura 3. API de integração entre o iRec e o OBP.

As configurações para utilização da ferramenta são compostas por vários arquivos *yaml* responsáveis por definir todos os parâmetros dos módulos. Um pesquisador pode definir os parâmetros do dataset, a pólitica que será usada em sua criação, os recomendadores para a simulação, e todas as métricas de avaliação offline e online, e outros. As Configurações 1, 2 e 3 ilustram como configurar os três módulos acima descritos.

Configuração 1. syntheticData.yaml

```
SyntheticData:
n_actions: 1000
dim_context: 23
reward_type: continuos
reward_function: linear
min_reward: 1
max_reward: 5
n_users: 943
users_context_file: path
users_frequency_file: path
obp_parameters: None
splitting:
  strategy: temporal
  train_size: 0.8
  test_consumes: 5
```

Configuração 2. experimentalSetup.yaml

```
ExperimentalSetup:
  irec_parameters_file: path
  input_file: path
  output_file: path
```

Configuração 3. evaluationSetup.yaml

```
EvaluationSetup:
  irec_parameters_file: "path"
  ope_estimators: [IPW,DM,DR]
  regression_model: linear
```

4. Experimentos

Nessa seção apresentamos uma avaliação experimental demonstrando as potencialidades e riquezas de análises que podem ser realizadas quando os dois principais frameworks de avaliação de recomendadores interativos são combinados. Utilizando nossa ferramenta de integração do iRec e OBP, dividimos nossos experimentos entre a avaliação tradicional e a contrafactual, contrapondo os resultados obtidos.

4.1. Avaliação Tradicional

Em nossos experimentos, primeiramente selecionamos três conjuntos de dados disponibilizados pelo *iRec*: *Netflix* (Filmes), *Good Books* (Livros) e *Yahoo Music* (Música). Comparamos o desempenho dos seguintes algoritmos, previamente descritos na Seção 2.1 e disponibilizados no *iRec*: e-Greedy, UCB, TS, Linear e-Greedy e Linear UCB, considerando três métricas implementadas pelo *iRec*: **Hits**: o números de recomendações que acertou o histórico do usuário (relevância); **ILD**: medida pela correlação de Pearson dos vetores de atributos dos itens entre a lista de itens recomendados (diversidade); e **Users Coverage**: representada pela porcentagem de usuários distintos que possuem relação com os itens recomendados (cobertura).

Dataset	Yahoo Music			Netflix			Good Books		
Métrica	Hits			Hits			Hits		
T	10	50	100	10	50	100	10	50	100
e-Greedy	1.460	7.424	13.360	1.320	5.390	9.936	0.800	3.072	5.633
UCB	1.358	7.330	13.277	1.284	5.440	9.915	0.764	2.984	5.493
TS	1.907	8.356	14.720	1.882	7.498	12.959	1.361	4.528	7.216
Linear e-Greedy	0.011	0.316	1.059	0.158	2.303	6.037	0.060	0.815	2.543
Linear UCB	**3.157▲**	**15.514▲**	**25.361▲**	**1.980▲**	**12.076▲**	**22.361▲**	**1.586▲**	**6.848▲**	**12.593▲**
Métrica	ILD			ILD			ILD		
T	10	50	100	10	50	100	10	50	100
e-Greedy	**0.461●**	0.462	0.465	0.401	0.416	0.421	0.489	0.494	0.495
UCB	**0.465●**	0.463	0.465	0.404	0.416	0.421	**0.490▲**	**0.495▲**	**0.495▲**
TS	0.431	0.452	0.459	0.336	0.373	0.387	0.467	0.487	0.492
Linear e-Greedy	**0.466▲**	**0.478▲**	**0.488▲**	**0.481▲**	**0.482▲**	**0.481▲**	0.487	0.488	0.488
Linear UCB	0.387	0.418	0.436	0.375	0.387	0.394	0.428	0.469	0.476
Métrica	UsersCoverage			UsersCoverage			UsersCoverage		
T	10	50	100	10	50	100	10	50	100
e-Greedy	0.684	0.960	0.985	0.545	0.857	0.954	0.499	0.853	0.932
UCB	0.664	0.960	0.986	0.544	0.872	0.949	0.483	0.842	0.930
TS	0.748	0.960	0.986	**0.631▲**	0.895	0.957	**0.636▲**	**0.893▲**	**0.944▲**
Linear e-Greedy	0.005	0.038	0.075	0.078	0.129	0.244	0.037	0.181	0.306
Linear UCB	**0.806▲**	**0.967▲**	**0.990▲**	0.588	**0.920▲**	**0.963▲**	0.489	0.819	0.915

Tabela 1. Performance dos modelos *bandit* no cenário offline. Os resultados foram comparados com o teste de Wilcoxon com $p\text{-}value = 0.05$. O símbolo ▲ denota ganhos estatísticos e o símbolo ● representa empates.

A Tabela 1, gerada automaticamente pelo *iRec*, mostra os resultados experimentais gerados com as configurações descritas anteriormente. Analisando os resultados, é importante observar que nenhum deles obteve os melhores resultados para todas as métricas avaliadas, demonstrando a importância de utilizar diferentes métricas de avaliação adequadas ao cenário de recomendação. Os SsR devem ser capazes de fornecer itens relevantes, diversos e novos, suprindo as necessidades de consumo da maioria dos usuários. Em termos de relevância (Hits), fica clara a superioridade do Linear UCB, independente do tamanho da lista recomendada. Considerando a diversidade, observamos que o Linear e-Greedy apresenta os melhores resultados para as coleções Yahoo Music e Netflix, enquanto o UCB foi superior para a coleção Good Books. Todavia, ao contrário da perspectiva de

relevância, a superioridade desses algoritmos não é tão discrepante sob a perspectiva de diversidade, reforçando a necessidade de testes estatísticos como o utilizado pelo *iRec* (i.e. Wilcoxon). Sob a perspectiva da cobertura, novamente observamos uma superioridade do Linear UCB. Porém, nesse caso, temos o TS com um desempenho bem próximo ao Linear UCB, chegando inclusive a ser estatisticamente superior na coleção Good Books.

4.2. Avaliação Contrafactual

Uma avaliação contrafactual permite avaliar SsR sem o viés de seleção e exposição. A partir de coleções pré-existentes, estima-se como seria o desempenho de uma nova estratégia caso fosse utilizada em substituição à política de origem, por meio dos estimadores. Os estimadores contrafactuais atuais exigem que conheçamos a política de produção usada para criar o conjunto de dados - que não está disponível para conjuntos de dados offline. Nesse sentido, utilizando o OBP, criamos um conjunto de dados de recomendação sintético para produzir classificações de 1 a 5, seguindo o mesmo padrão como o tradicional MovieLens 100k (100 mil avaliações para a mesma quantidade de usuários). Basicamente, cada contexto contém o ID do usuário e suas características (e.g., gênero, idade e ocupação). As recomendações são feitas por uma política Linear. Nesse caso, avaliamos os mesmos SsR do experimento anterior, possível somente após a integração do *iRec* com o OBP. Eles foram configurados para realizar 100 recomendações para cada usuário. Cada SR é avaliado por três estimadores distintos, disponibilizados pelo OBP e descritos na Seção 2.2 (i.e. IPS, DM e DR).

Dataset	Synthetic Dataset								
Measure	Estimated policy value			95.0% CI (lower) – 95.0% CI (upper)			Relative policy value		
Estimators	IPS	DM	DR	IPS	DM	DR	IPS	DM	DR
e-Greedy	6.244	4.468	4.495	2.854 - 10.154	4.458 - 4.477	4.432 - 4.596	1.901	1.36	1.368
UCB	8.318	4.484	4.438	3.821 - 13.279	4.474 - 4.492	4.397 - 4.473	**2.532**	1.365	1.351
TS	3.248	4.303	4.312	1.511 - 5.274	4.294 - 4.312	4.273 - 4.367	0.988	1.31	1.312
Linear e-Greedy	2.874	3.973	3.916	1.086 - 5.133	3.945 - 4.0	3.828 - 3.981	0.875	1.209	1.192
Linear UCB	3.35	4.893	4.89	1.192 - 5.952	4.89 - 4.895	4.864 - 4.918	1.02	**1.489**	**1.488**

Tabela 2. Estimadores Contrafactuais para os modelos bandits nos dados sintéticos. O DR é o estimador mais imparcial e consistente para garantir uma avaliação confiável. Os melhores resultados estão em negrito.

Os resultados são destacados na Tabela 2. A primeira e a segunda colunas referem-se ao valor médio estimado da política e ao intervalo de confiança feito pelas métricas selecionadas para cada recomendador. O DM é influenciado pela política original (Linear) usada para criar o conjunto de dados, o IPS apresenta uma variância maior do que os outros métodos. DR é o valor mais imparcial e consistente. A terceira coluna apresenta o valor relativo da política – o quanto esta nova política melhorou a política original. Valores superiores a 1 significam que a política teria um desempenho melhor se tivesse substituído a política original. Avaliando os resultados, observamos que o Linear UCB e o UCB são as estratégias que apresentam os melhores desempenhos, reforçando os resultados obtidos na avaliação tradicional. Todavia, o TS, que foi destaque anteriormente, na avaliação contrafactual apresentou um desempenho abaixo do esperado, não sendo capaz, em alguns casos, de sequer superar o recomendador original (linear). Isso demonstra como o TS, de certa forma, se beneficia do viés pré-existente nas coleções pré-existentes.

5. Conclusão e Trabalhos Futuros

Esse trabalho preencheu uma lacuna existente na literatura referente à avaliação de Sistemas de Recomendação (SsR) interativos baseados em abordagens Multi-Armed

Bandits MAB por meio da integração de dois dos mais importantes frameworks de avaliação: *iRec* e Open Bandit Pipeline. O primeiro está focado em avaliações offline, disponibilizando um conjunto completo de coleções, métricas e SsR. O segundo oferece metodologias e métricas focadas em avaliações contrafactuais. Nossa integração permite que qualquer solução bandit possa ser avaliada de forma offline e sob a perspectiva contrafactual, considerando, inclusive, coleções sintéticas, livres do viés das coleções pré-selecionadas. Realizamos uma avaliação da integração realizada considerando 4 coleções (3 selecionadas e 1 sintética), 6 métricas de avaliação (3 offline e 3 contrafactuais) e quatro algoritmos estado-da-arte em MAB. Nossa primeira observação é que para dizer que um modelo é melhor que outro, não basta avaliar apenas uma métrica e os resultados a curto, médio ou longo prazo. Outra observação importante em nossos resultados diz respeito ao quanto alguns algoritmos se "beneficiam" do viés dos dados, levando a conclusões equivocadas de seu desempenho. Nesses casos, a efetividade de um modelo pode estar muito mais associada ao quanto ele consegue se aproximar das recomendações realizadas pelo recomendador que originou a coleção do que efetivamente satisfazer o usuário. Como trabalho futuro, pretendemos realizar uma avaliação ainda mais ampla, considerando muitos outros modelos MAB, métricas e coleções de dados.

Referências

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256.

Liu, Y., Yen, J.-N., Yuan, B., Shi, R., Yan, P., and Lin, C.-J. (2022). Practical counterfactual policy learning for top-k recommendations. In *ACM SIGKDD*, pages 1141–1151.

Pan, W., Cui, S., Wen, H., Chen, K., Zhang, C., and Wang, F. (2021). Correcting the user feedback-loop bias for recommendation systems. *arXiv preprint arXiv:2109.06037*.

Saito, Y., Aihara, S., Matsutani, M., and Narita, Y. (2020). Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. *arXiv preprint arXiv:2008.07146*.

Sanz-Cruzado, J., Castells, P., and López, E. (2019). A simple multi-armed nearest-neighbor bandit for interactive recommendation. In *RecSys*, pages 358–362.

Shams, S., Anderson, D., and Leith, D. (2021). Cluster-based bandits: Fast cold-start for recommender system new users.

Silva, T., Silva, N., Werneck, H., Mito, C., Pereira, A. C., and Rocha, L. (2022). irec: An interactive recommendation framework. In *SIGIR*, pages 3165–3175.

Wu, Q., Iyer, N., and Wang, H. (2018). Learning contextual bandits in a non-stationary environment. In *SIGIR*, pages 495–504.

Yang, Y., Xia, X., Lo, D., and Grundy, J. (2022). A survey on deep learning for software engineering. *ACM Computing Surveys (CSUR)*, 54(10s):1–73.

Zhou, S., Dai, X., Chen, H., Zhang, W., Ren, K., Tang, R., He, X., and Yu, Y. (2020). Interactive recommender system via knowledge graph-enhanced reinforcement learning. In *SIGIR*, pages 179–188.