

# Recomendação de Jogos Calibrada

Giovanni Ferreira Martinelli  
Universidade Federal de Minas Gerais  
Belo Horizonte, Minas Gerais

Gabriel Soares da Silva  
Universidade Federal de Minas Gerais  
Belo Horizonte, Minas Gerais

## 1 INTRODUÇÃO

Atualmente existe uma grande quantidade de opções de jogos disponíveis em lojas online. A oferta é tamanha que, para que o usuário não se sinta perdido, é necessário sistemas de recomendações para guiar o usuário em uma exploração que faça sentido para ele. Esses recomendadores, de modo geral, são treinados de forma a otimizar métricas de avaliação que nem sempre vão em direção aos interesses do usuário. Em alguns casos, eles podem acabar por ocultar itens que seriam de interesse do usuários.

Diante desse cenário, nosso trabalho consiste em fazer recomendações calibradas no domínio de jogos, utilizando o conceito de "justiça" (*fairness*) ao invés de acurácia, comumente utilizada em recomendações. Esse conceito foi primeiramente explorado por Harald Stack em 2018 [1], no domínio de filmes, sendo esse o estado da arte para esse modelo de recomendação.

Nosso trabalho visa explorar as diferenças no domínio entre jogos e filmes, analisando a relevância desse modelo para o contexto de jogos. Visávamos explorar as diferenças temporais advindas com o novo domínio, porém devido a base de dados utilizada não foi possível explorar esse contexto.

## 2 IMPLEMENTAÇÃO

Nossa implementação deste trabalho pode ser encontrada em: <https://github.com/GiowGiow/Calibrated-Game-Recommender>

## 3 TRABALHOS RELACIONADOS

Esse modelo de calibragem utilizando *fairness* no contexto de Sistemas de Recomendação foi primeiramente introduzido por Harald Stack na RecSys 2018. Desde então não houveram publicações estendendo esse modelo.

Para o recomendador utilizamos um modelo Bayesiano de fatoração de matrizes, similarmente visto em vários recomendadores para jogos da Steam.

## 4 SOLUÇÃO PROPOSTA

### 4.1 Base de dados

O dataset utilizado para nosso trabalho foi o "*Steam Video Game and Bundle Data*" disponibilizado em: [https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam\\_data](https://cseweb.ucsd.edu/~jmcauley/datasets.html#steam_data)

### 4.2 Modelagem

Nossa modelagem para o domínio de jogos teve que alterar importantes aspectos do trabalho original. Utilizamos somente do *feedback* implícito para fazer as recomendações, já que o *feedback* explícito não estava disponível.

Baseado nas observações que fizemos no *dataset*, a maior parte dos jogadores joga menos da metade dos jogos que possuem. Isso mostra que eles tem interesse nos jogos que compraram, porém

possuem uma preferência para os jogos que eles jogaram. Adicionalmente, os jogadores geralmente gastam um tempo significativo a mais jogando seu jogo preferido do que os outros jogos que possuem.

Dado esses fatores utilizamos as horas jogadas de cada jogo por cada usuário o indicador de quanto ele gosta de um jogo. Através desse tempo de jogo formulamos uma matriz de suas preferências.

Nós definimos o tempo de jogo em uma matriz  $T$ , em que cada entrada  $T_{ij}$  representa o tempo jogado em minutos que o usuário  $i$  gastou no jogo  $j$ . Nós criamos uma aproximação de uma matriz de probabilidade  $S$ , que representa a matriz de preferência do usuário. Um jogador jogaria um jogo que tivesse as seguintes propriedades:

- Uma função que mapeia os valores de  $T_{ij} \in (0, \infty)$  para  $S_{ij} \in (0, 1)$
- A probabilidade de  $S_{ij}$  aumenta com o tempo de jogo  $T_{ij}$
- $S_{ij}$  e  $T_{ij}$  tem uma relação não linear.  $S_{ij}$  deve aumentar marginalmente quando  $T_{ij}$  aumentar

De acordo com essas necessidades nós usamos a função *tanh* para criar uma nova matriz com esses valores. Nós limitamos o número máximo de horas de qualquer jogo para 50 horas, e baseado nas nossas observações que  $\tanh(2) = 1$ , nós construímos a matriz de probabilidade tal que:

$$S_{ij} = \tanh\left(\frac{T_{ij} * 2}{50 * 60}\right)$$

Nós fizemos várias tarefas para preparar o dataset para treino. Primeiro filtramos os jogos que nunca foram jogados por nenhum usuário. Depois, devido a restrições de memória, limitamos o número de jogos que investigamos. Limitando o ID que a Steam usa para os jogos para todos abaixo de 10000. Como resultado 354 jogos foram considerados.

Para cada usuário, extraímos uma lista de jogos que ele jogou, com os nomes dos mesmos e seus gêneros.

### 4.3 Recomendador

Como visto por nós sobre o estado da arte para recomendação de jogos, utilizamos o recomendador de ranqueamento personalizado Bayesiano de fatoração de matrizes para feedback implícito. No nosso caso utilizando o da biblioteca *Implicit* para Python. Particionamos 80% do dataset para o treinamento e 20% pra testes. A diversidade é incluída pelo recomendador e não no processo de reranqueamento.

### 4.4 Calibragem

Aqui explicaremos as métricas que quantificam o grau de calibração de uma lista de jogos recomendados, com relação ao histórico de jogos jogados pelo usuário. Para este fim, consideramos duas distribuições, ambas baseadas na distribuição de gêneros  $g$  para cada jogo  $i$ , dado por  $p(g|i)$ :

- $p(g|u)$ : A distribuição de gêneros  $g$  do set de jogos  $H$  jogados pelo usuário  $u$  no passado.

$$p(g|u) = \sum_{i \in H} p(g|i)$$

- $q(g|u)$ : A distribuição de gêneros  $g$  da lista de jogos recomendados para o usuário  $u$

$$q(g|u) = \sum_{i \in I} p(g|i)$$

Em que  $I$  é o set de jogos recomendados.

#### 4.5 Métrica de calibragem

O método escolhido para comparar o quão similares são duas distribuições uma com a outra foi a divergência de Kullback-Leibler

$$C_{k,l} = \sum p(g|u) * \log \frac{p(g|u)}{\tilde{q}(g|u)}$$

Tendo essa definição de  $\tilde{q}(g|u)$  para o caso em que seja 0:

$$\tilde{q}(g|u) = (1 - \alpha) * q(g|u) = \alpha * p(g|u)$$

Com um  $\alpha$  pequeno como 0.01.

#### 4.6 Recomendação Calibrada

re-ranking the predicted list of a recommender system in a post-processing step

A recomendação calibrada é feita na etapa do pós-processamento, rerranqueando a lista dada pelo recomendador. Para determinar um set ótimo  $I^*$  de  $N$  itens recomendados, utilizamos a máxima relevância marginal, dada por:

$$I^* = \underset{I, |I|=N}{\operatorname{argmax}} (1 - \lambda) * s(i) - \lambda * C_{k,l}(p, q(I))$$

Em que:

- $\lambda \in (0, 1)$  determina o trade-off da calibragem
- $s(i)$  é o score dos itens  $i \in I$  preditos pelo recomendador e  $s(I) = \sum_{i \in I} s(i)$

### 5 RESULTADOS

Ao testar no dataset escolhido percebemos a predominância do gênero *Action* nos jogos, aparecendo predominantemente nas recomendações. Porém devido a esse cenário, outros jogos relevantes para o usuário que tinham uma pequena proporção nos gêneros do usuário eram comumente desconsiderados nas listas Top N, dado que o recomendador preza por acurácia e não justiça.

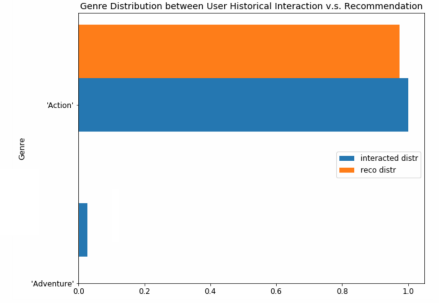
Essa situação foi oportuna para testar a recomendação calibrada, já que ao rerranquear a lista dada, esses gêneros de jogos mal representados voltavam a aparecer. Provando que a recomendação calibrada no domínio de jogos, que possui grande discrepância no tempo de jogo entre gêneros, uma opção viável para fazer uma recomendação justa de acordo com as preferências do usuário.

Um problema desse pós-processamento é perder acurácia ao rerranquear a lista, por isso é necessário achar o melhor  $\lambda$  para o dataset em questão.

Aqui temos um teste exemplo de uma recomendação calibrada e outra descalibrada:

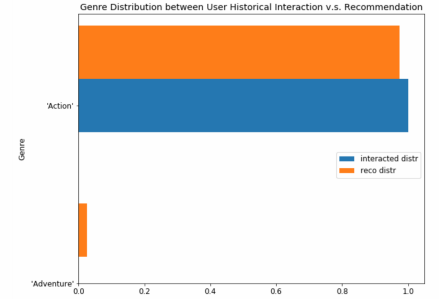
Não Calibrado:

- Usuário:
- 0.955 ação
  - 0.045 aventura



Calibrado:

- Usuário:
- 0.955 ação
  - 0.045 aventura



O pós processamento teve comportamentos erráticos com alguns usuários, não soubemos dizer se foi por problemas na implementação.

### 6 CONCLUSÃO

O calibragem de uma lista recomendada pode ser proveitosa em vários contextos em que a fidelidade aos gostos do usuário deve ser prioridade, no domínio dos jogos essa proposta é muito relevante e impede o overfit dos recomendadores focados em acurácia. Dando ao jogador recomendações corretas baseadas nos jogos que ele já jogou, mesmo que a distribuição de gêneros seja bastante discrepante.

### REFERENCES

- [1] Harald Steck. 2018. Calibrated Recommendations. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys '18)*. ACM, New York, NY, USA, 154–162. <https://doi.org/10.1145/3240323.3240372>