Sistemas de Recomendação

Material Didático sobre Sistemas de Recomendação Sequenciais

Gustavo Wadas Lopes - 12745640
Lívia Lelis - 12543822

Novembro de 2024
ICMC/USP

Introdução

Esse projeto tem como propósito a produção de um material didático sobre sistemas de recomendação sequenciais. O objetivo é apresentar uma visão geral sobre o assunto, enquanto apresentamos uma abordagem de implementação de sistemas de recomendação sequenciais baseada em redes neurais de atenção e um modelo baseado em fatoração de matrizes com cadeias de Markov.

Metodologia

Para esse projeto, iremos utilizar dois algoritmos de recomendação: o Self-Attentive Sequential Recommendation (SASRec) [3] e o Factorizing Personalized Markov Chains (FPMC) [5]. O SASRec é um algoritmo proposto por Wang-Cheng Kang e Julian McAuley em 2018 consiste em uma rede neural de atenção que é treinada para prever o próximo item a ser recomendado em uma recomendação sequencial. Já o FPMC é um algoritmo proposto por Rendle, Freudenthaler e Schmidt-Thieme em 2010 e consiste de uma abordagem híbrida baseando-se na ideia de fatoração de matrizes e cadeias de markov junto a técnica BPR (Bayesian Personalized Ranking) para prever a próxima ”cesta”(uma única música no nosso contexto) a ser recomendada a partir da ”cesta”anterior.

Além disso, iremos utilizar o dataset de reproduções de músicas do site last.fm disponível em http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html [1] por Celma [2].

Esse dataset de 2010 apresenta o histórico de reprodução de músicas de uma amostra de 1.000 usuários. Pela natureza do last.fm, cada usuário tem em média aproximadamente 20 mil reproduções de músicas, resultando em um total de aproximadamente 19 milhões de reproduções de músicas no dataset.

Produzimos dois notebooks disponíveis em https://github.com/LiviaLelis/recsys-tp-seq, junto ao material textual do projeto e instruções de uso. Todo o material e explicações estão disponíveis pelos próprios notebooks.

Além disso, foi produzida uma vídeo aula explicativa sobre sistemas de recomendação sequencial disponível em https://www.youtube.com/watch?v=rnXUyPji_-o, para dar uma breve introdução ao assunto e apresentar as implementações feitas neste projeto.

O público-alvo deste material é principalmente pessoas que atuem na área de computação, com conhecimento técnico e interesse em aprender sobre sistemas de recomendação, aprendizado de máquina e aplicações práticas.

Implementação

Dataset

Para o dataset, fizemos algumas modificações para adequar-se as necessidades do algoritmo. Primeiramente, fizemos um mapeamento dos UUIDs (universally unique identifier) para inteiros sequenciais a partir do 1, já que necessitamos de representações numéricas para os algoritmos.

Além disso, devido à natureza do last.fm, cada usuário tem um volume extenso de reproduções de músicas. Entendemos que esse histórico extenso pode ser dívido em "sessões de reprodução", isto é, um conjunto de reproduções de músicas que ocorrem de maneira quase contínua no tempo. Portanto, para cada usuário, construímos um conjunto de sessões, tomando como critério para uma sessão a reprodução de músicas que ocorrem em um intervalo máximo de 30 minutos entre duas reproduções consecutivas. Acreditamos que essa abordagem de tratamento de dados seja mais adequada para o problema, reduzindo o tamanho das sequências e mantendo-as mais coerentes, já que as preferências musicais de um usuário pode variar entre sessões e com o tempo, entretanto não conseguimos medir se essa assunção é verdadeira.

Modelo de Atenção

O SASRec [3] é um modelo lançado em 2018 e foi um dos primeiros algoritmos de recomendação sequencial baseado na técnica de self-attention introduzida introduzida no artigo Attention Is All You Need [7], sendo o SASRec um dos pioneiros trazendo essa ideia para o campo de sistemas de recomendação sequenciais.

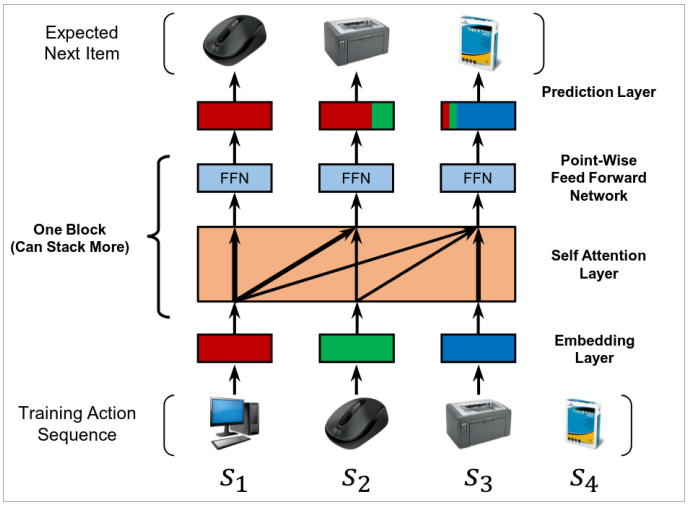


Figura 1: Diagrama da arquitetura do modelo SASRec [3]

Nossa implementação é baseada em uma adaptação [9] em PyTorch do modelo dos autores originais implementado em TensorFlow. Nessa adaptação, não fizemos mudanças diretas na arquitetura do modelo, mas apenas algumas adaptações de código para melhorar a legibilidade e o desempenho. Além disso, desviamos da implementação original nos parâmetros escolhidos, otimizador e outras configurações.

A versão final do modelo foi treinada com uma amostra aleatória de 100 mil sessões para reduzir o tempo de treinamento ao custo de uma perda potencialmente pequena de qualidade. Além disso, para a divisão dos dados, fizemos uma divisão de treino, teste e validação pela estratégia sugerida pelos autores de truncamento das sessões, isto é, todas as divisões contêm todas as sessões, sendo que na divisão de treino é descartada os dois últimos itens da sessão, na de validação o último item da sessão e na de teste é mantida a sessão inteira. Como nosso volume de dados era suficientemente grande, acreditamos

que poderíamos obter resultados mais realistas se seguíssemos uma divisão por meio das sessões em si ao invés deste truncamento.

Além disso, para melhorar a performance do modelo e viabilizar o treinamento, é adotada a estratégia original de construção de amostras negativas, isto é, geramos para cada item um conjunto de itens negativos para que o treinamento seja mais eficiente, realizando uma classificação binária de se o item é positivo ou negativo.

A função de perda utilizada para avaliar o modelo foi a Binary Cross Entropy with Logits, que é uma função de perda para classificação binária. Além disso fizemos uso do algoritmo de otimização AdamW [4] com o parâmetro de learning rate fixo. Também fizemos experimentos com o scheduler OneCycleLR [6] que apresentou resultados bem semelhantes, por isso seguimos com a learning rate fixa para deixar os resultados mais comparáveis.

Mais informações sobre os parâmetros escolhidos podem ser vistas no repositório do projeto disponível em https://github.com/LiviaLelis/recsys-tp-seq.

FPMC

O FPMC [5] é um modelo lançado em 2010 que combina a técnica de fatoração de matrizes (usando BPR) com a técnica de cadeias de Markov.

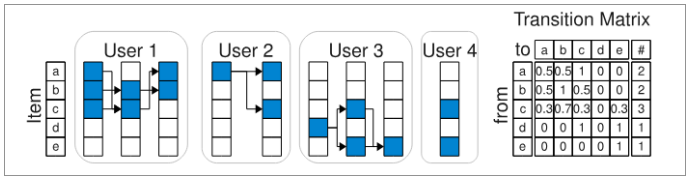


Figura 2: Diagrama da arquitetura do modelo FPMC [5]

A implementação do modelo FPMC é comparativamente simples, sendo construído a partir da combinação de duas componentes: a fatoração matricial representando as preferências dos usuários e características dos itens por meio de vetores latentes; e a cadeia de Markov que modela as dependências sequenciais entre itens consecutivos. O modelo é treinado por meio da abordagem BPR (Bayesian Personalized Ranking), que é uma técnica de aprendizado de ranking muito utilizada para otimizar modelos baseados em vetores latentes (como fatoração de matricial isoladamente).

Além disso, a estrutura dos dados consumidos pelo modelo consiste de tuplas do tipo (usuário, item anterior, próximo item real, próximo item negativo [gerado no dataset]), ou seja, diferentemente do modelo SASRec, o usuário é levado em consideração, mas, ao invés de tomar como entrada uma sequência de itens, recebe apenas o item anterior.

Resultados

A fim de melhor comparar os dois modelos, optamos por em ambos utilizar a mesma amostra de 100k sessões para reduzir o tempo de treinamento ao custo de uma redução na qualidade da base de dados. Além disso, as condições de treinamento foram definidas mantendo parâmetros semelhantes: ambos com um batch size de 1024 e sem scheduler. Além disso, foi selecionado um número de epochs para manter ambos tempos de treinamento próximos (aproximadamente 40 minutos em uma RTX 3080TI).

Modelo	NDCG@10	HIT@10
SASRec	0.6757	0.7704
FPMC	0.6733	0.7437

Tabela 1: Métricas de qualidade do SASRec e FPMC nos dados de teste

Ambos os modelos apresentam resultados muito parecidos, com o SASRec tomando uma leve margem de melhoria. Ambos os modelos ainda apresentam potencial de melhoria já que nenhum deles paracia ter convergido completamente durante o treinamento, além da possibilidade de tunar os hiperparâmetros de treinamento.

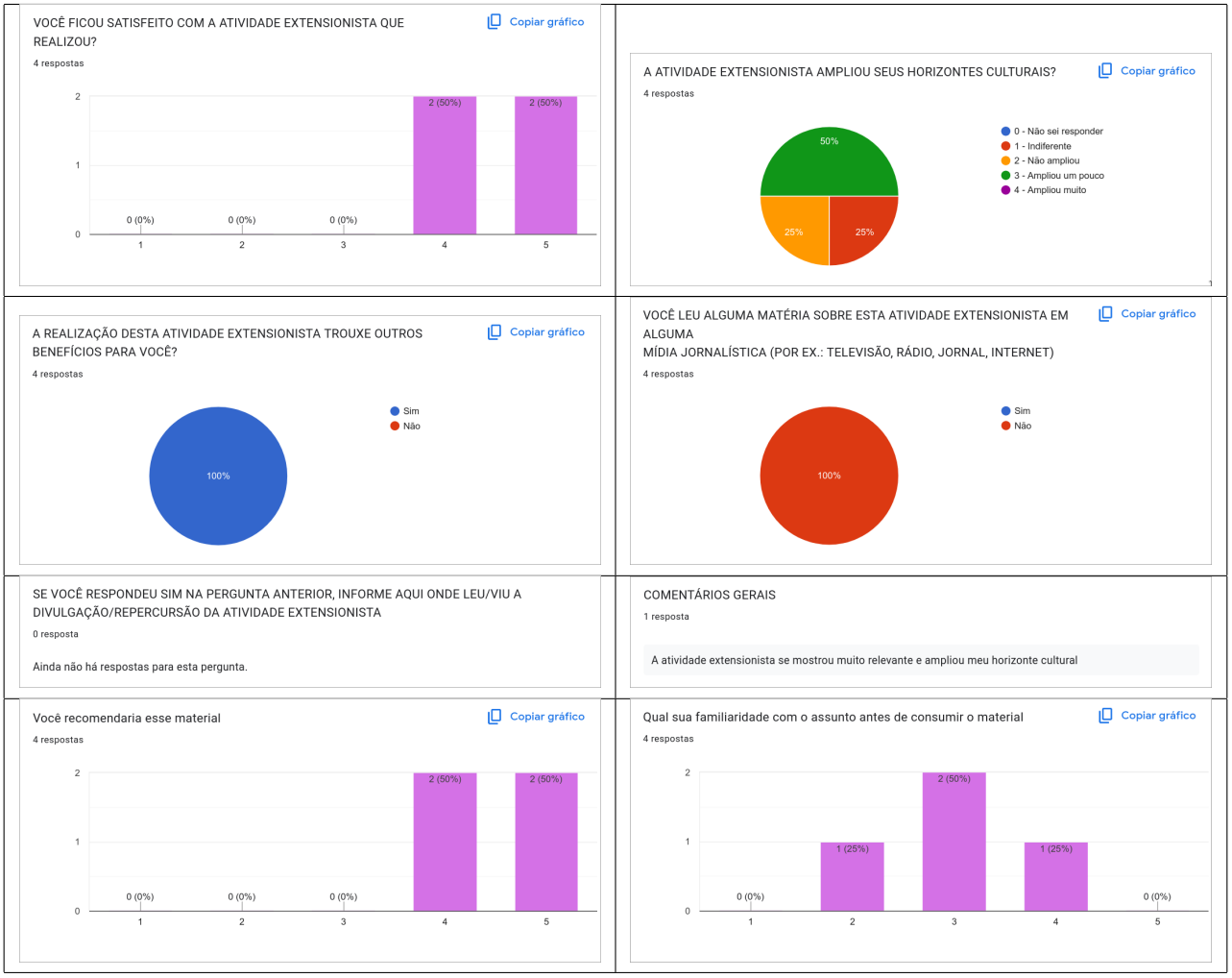
Desse modo, não é possível declarar um modelo como claramente superior ao outro, porém o FPMC costuma ser menos custoso e convergir mais rapidamente que o SASRec, sendo uma boa alternativa para situações onde é necessário reduzir o custo de treinamento ao mesmo tempo que se mantém um bom resultado.

No artigo original do modelo SASRec [3], ele supera o FPMC em todos os datasets utilizados (que não incluem o utilizado aqui), mas também em uma margem de melhoria moderada (mas mais significativa do que as apresentadas aqui).

Avaliação do material

Para a avaliação do material foi utilizado o Google Forms seguindo o formato sugerido na ”Regulamentação da Curricularização da Extensão na Universidade de São Paulo“. O formulário juntamente ao material em vídeo e o repositório foram enviados para um grupo de ex-alunos e alunos da área de computação sem vínculo ativo ou anterior à Universidade de São Paulo, para coleta de feedbacks sobre o material e a atividade extensionista.

Algumas das perguntas necessárias não fazem muito sentido para o projeto, entretanto o feedback geral foi positivo, por mais que o número de participantes tenha sido bem pequeno, tendo apenas 4 respostas.



Referências

[1] O. Celma. Last.fm 1K Users Dataset. 2010. URL: http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html.

[2] O. Celma. Music Recommendation and Discovery in the Long Tail. Springer, 2010.

[3] Wang-Cheng Kang e Julian McAuley. Self-Attentive Sequential Recommendation. 2018. arXiv: 1808.09781 [cs.IR]. URL: https://arxiv.org/abs/1808.09781.

[4] Ilya Loshchilov e Frank Hutter. Decoupled Weight Decay Regularization. 2019. arXiv: 1711.05101 [cs.LG]. URL: https://arxiv.org/abs/1711.05101.

[5] Steffen Rendle, Christoph Freudenthaler e Lars Schmidt-Thieme. "Factorizing personalized Markov chains for next-basket recommendation". Em: Proceedings of the 19th International Conference on World Wide Web. WWW '10. Raleigh, North Carolina, USA: Association for Computing Machinery, 2010, pp. 811–820. ISBN: 9781605587998. DOI: 10.1145/1772690.1772773. URL: https://doi.org/10.1145/1772690.1772773.

[6] Leslie N. Smith e Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. 2018. arXiv: 1708.07120 [cs.LG]. URL: https://arxiv.org/abs/1708.07120.

[7] Ashish Vaswani et al. Attention Is All You Need. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[8] Shoujin Wang et al. "Sequential/Session-based Recommendations: Challenges, Approaches, Applications and Opportunities". Em: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '22. ACM, jul. de 2022, pp. 3425–3428. DOI: 10.1145/3477495.3532685. URL: http://dx.doi.org/10.1145/3477495.3532685.

[9] Sean (Seok-Won) Yi. PyTorch implementation for SASRec. 2023. URL: https://github.com/seanswyi/sasrec-pytorch.