Recomendação baseada em itens utilizando similaridade entre node-embeddings

VINÍCIUS SILVA, Universidade Federal de Minas Gerais, Brasil

ACM Reference Format:
Vinícius Silva. 2021. Recomendação baseada em itens utilizando similaridade entre node-embeddings. 1, 1 (April 2021), 3 pages.

1 INTRODUÇÃO

É sabido que sistemas de recomendação são ubíquos em sites de comércio eletrônico, uma vez que estes disponibilizam milhares de itens a venda e gerar recomendações personalizadas de produtos potencialmente interessantes para cada cliente é de grande importância estratégica. Uma das abordagens mais comuns para se projetar sistemas desse tipo é construir um mecanismo capaz de quantificar a similaridade entre qualquer par de itens (i, j) e utilizá-lo para prever quão similar é um item candidato c aos itens previamente consumidos e avaliados por um usuário u.

O principal desafio dentro desse contexto está no fato de que para que seja possível quantificar as similaridades entre itens, é necessário que os representemos matematicamente na forma de vetores. A intuição é que vetores $\vec{v}_i$ e $\vec{v}_j$ devem ser próximos no espaço vetorial se os itens i e j são altamente similares. Várias abordagens para esse problema são possíveis e este trabalho tem o objetivo de explorar técnicas de Machine Learning que geram tais representações automaticamente através da otimização de uma função de perda (*loss function*).

Os algoritmos considerados estado da arte para se resolver este problema baseiam-se em técnicas de aprendizado profundo [5], em que redes neurais artificiais de múltiplas camadas são utilizadas para aprender representações que fazem sentido para cada item em um espaço vetorial latente. Técnicas deste tipo são normalmente bastante custosas do ponto de vista computacional e estão além do escopo deste curso, e portanto, trataremos apenas de técnicas consideradas *shallow*. Considerando este contexto, podemos destacar o algoritmo Item2Vec [1], que é uma adaptação feita no famoso algoritmo Word2Vec [4] para lidar com itens em um catálogo ao invés de palavras em um vocabulário.

2 SOLUÇÃO

A ideia central deste trabalho é utilizar o conhecimento adquirido em recomendações **não**-personalizadas para gerar recomendações personalizadas. Além da matriz de interação, que contém as notas que os usuários deram a itens consumidos, também utilizamos dados provenientes de um *crawler* que indicam padrões frequentes detectados pela Amazon no comportamento de seus clientes (ex.: "clientes que compraram x também compraram y") e nosso objetivo é levar em conta tais padrões para medir a similaridade entre produtos.

Nossa abordagem compartilha da mesma intuição por trás do Item2Vec, mas possui algumas diferenças que merecem ser destacadas. Em suma, o Item2Vec é um *sequence model* que considera como sequência uma lista de produtos avaliados por um mesmo usuário e utiliza uma rede neural *shallow* para prever qual o próximo item desta sequência. Já a abordagem proposta, baseada em Node2Vec [2], também é um *sequence-model* baseado em uma rede neural *shallow*, mas que considera como sequência, os itens visitados em um passeio aleatório em um grafo de similaridades pré-existente.

A limitação do algoritmo Item2Vec que queremos explorar ao realizar este trabalho está justamente no fato de que ele não permite que conhecimento prévio a respeito das relações entre itens seja levado em consideração no aprendizado dos embeddings. Muitas vezes, é possível obter este conhecimento (através de técnicas de mineração de padrões frequentes, por exemplo), mas algoritmos que se baseiam apenas na matriz de interação entre usuários e itens não são capazes de levá-lo em conta.

Author's address: Vinícius Silva, viniciusoliveira@dcc.ufmg.br, Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brasil.

Em suma, utilizamos o algoritmo Node2Vec para gerar embeddings para cada produto presente no conjunto de treino e então medimos a similaridade entre itens a partir dessas representações. Uma vez que as similaridades são calculadas, utilizamos *weighted-sum* para gerar previsões das notas que um usuário u ele daria a um determinado item i caso u avaliasse i.

3 AVALIAÇÃO EXPERIMENTAL

Para analisarmos a performance da nossa solução, dividimos a matriz de interação temporalmente, de forma que os 85% das avaliações mais antigas são utilizadas para treino e os 15% mais recentes são utilizados para teste. Para se ter noção da qualidade dos resultados obtidos, comparamos nossa metodologia com 3 baselines:

(1) **Filtragem Colaborativa k-NN baseada em itens:** Este é o baseline mais simples dentre os utilizados. Consiste de um recomendador básico que representa os itens como um vetor de avaliações dadas pelos usuários que os consumiram. Conforme visto em aula, as predições são o resultado de uma *weighted-sum* em que as similaridades são calculadas através da distância de cosseno entre as representações dos itens.
(2) **Item2Vec:** Trata-se da aplicação do Word2Vec em um corpus formado por conjuntos de itens avaliados pelos clientes da Amazon. Para cada usuário u da base, geramos duas listas: Uma composta por itens avaliados positivamente ($rating >= 4$) e outra composta por itens avaliados negativamente ($rating < 4$). Essas listas compõem o corpus que é fornecido ao Word2Vec que interpreta cada produto como uma palavra e cada lista como uma frase. Ao final do treinamento, temos representações densas para cada item.
(3) **SVD++:** Técnicas baseadas em fatorização de matrizes são consideradas o estado da arte em filtragem colaborativa e por isso, decidimos comparar nossos resultados com os obtidos por um algoritmo desse tipo. Escolhemos utilizar o SVD++ especificamente por ser uma das técnicas mais comuns e por gerar bons resultados na prática. A implementação dessa técnica em nosso trabalho se deu através da biblioteca Python *Surprise* [3], que fornece implementações eficientes de vários algoritmos de recomendação.

4 RESULTADOS

Os resultados obtidos pela nossa abordagem são comparados com os baselines descritos na seção anterior e mostrados na Tabela 1. É importante salientar que, para computar essas métricas, removemos exemplos *cold-start* da base de teste, uma vez que não é possível aprender representações para itens que não estão na base de treinamento.

	Item2Vec	kNN Item-CF	SVD++	**Nossa abordagem**
Precision	0,833	0,835	0,810	**0,853**
Recall	0,911	0,925	0,965	**0,922**
Accuracy	0,786	0,799	0,794	**0,808**
RMSE	1,168	1,087	1,178	**1,108**
NDCG médio	0,961	0,936	0,966	**0,967**

Table 1. Comparativo entre a nossa abordagem e os baselines considerados.

5 CONCLUSÃO

Conforme mostram os resultados obtidos, podemos concluir que faz sentido a premissa do trabalho de que a inclusão de conhecimento prévio a respeito das similaridades entre os itens pode melhorar a qualidade de uma recomendação baseada em filtragem colaborativa. Esse resultado não surpreende, uma vez que é natural que modelos com acesso a mais dados tenham um melhor desempenho preditivo. Dessa forma, podemos dizer que a abordagem investigada por este trabalho pode ser uma boa alternativa para a personalização de recomendações quando as relações entre os itens disponíveis no catálogo estão disponíveis.

Como trabalhos futuros, seria interessante investigar como a nossa abordagem se comporta quando *sequence-models* mais avançados (como *Transformers*, por exemplo) são utilizados no aprendizado de representações, especialmente no caso em que as sequências que compõem o *corpus* estão ordenadas temporalmente, e não em ordem arbitrária como foi abordado neste trabalho.

REFERENCES

[1] Oren Barkan and Noam Koenigstein. 2016. Item2vec: neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[2] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[3] Nicolas Hug. 2020. Surprise: A Python library for recommender systems. *Journal of Open Source Software* 5, 52 (2020), 2174. https://doi.org/10.21105/joss.02174

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[5] Wen Zhang, Yuhang Du, Taketoshi Yoshida, and Ye Yang. 2019. DeepRec: A deep neural network approach to recommendation with item embedding and weighted loss function. *Information Sciences* 470 (2019), 121–140.