

Exploring temporal contexts for neighborhood-based models in Session-Based Recommender Systems

Gabriel B. Domingos
gbaltazar@usp.br
ICMC-USP
São Carlos, São Paulo

Matheus Takata
mtakata@usp.br
ICMC-USP
São Carlos, São Paulo

Igor Lovatto
igor.lovatto@usp.br
ICMC-USP
São Carlos, São Paulo

Marcelo Manzato
mmanzato@icmc.usp.br
ICMC-USP
São Carlos, São Paulo

RESUMO

Recent works in the field of Recommender Systems have focused in making recommendations using only information about accessed items in an active user session, since the user's preferences can be very specific to the context of the session. Despite the increasing interest in deep neural network models in the Recommender Systems in general, simpler models, such as neighborhood-based collaborative filtering ones, have been outperforming these more complex models in the session-based scenario. However, in order to soften known scalability problems in neighborhood-based models, sampling strategies during the recommendation process have been proposed, showing that some session contexts such as recency of neighbor sessions can improve the recommendation in some domains. Therefore this project explores the effect of different temporal contexts that can easily be captured from an anonymous user session, such as the hour of the day and day of the week, as alternatives to the established sampling methods.

KEYWORDS

session-based recommendation, contextual information, k-nearest neighbors

1 INTRODUÇÃO

Baseado na grande quantidade de informações presentes na Internet e outras plataformas atuais, houve um significativo aumento na quantidade de conteúdo disponível. A partir disso, é necessário que se tenha algum tipo de filtragem, ou seleção de conteúdo, para que a experiência de usuários, ou facilidade na visualização de conteúdo seja melhorada. Para isso, os sistemas de recomendação podem ser usados para filtrar conteúdos personalizados, recomendando itens de acordo com as preferências individuais de cada usuário [1].

A ideia principal dos sistemas de recomendação consiste em utilizar interações passadas do usuário para inferir suas preferências, e assim, recomendar itens que são relevantes de acordo com seus interesses. Tais informações podem ser fornecidas manualmente pelo indivíduo (como notas de avaliação, questionários e pesquisas),

até dados colhidos de forma implícita, como o histórico de navegação, cliques, *timestamps*, etc. Em particular, quando se trata de dados implícitos, existe uma dificuldade para inferência de algumas informações sobre os interesses do usuário, pois muitas vezes, tanto os metadados sobre os itens como o grau de satisfação dos indivíduos não estão prontamente disponíveis.

Uma das técnicas mais tradicionais e conhecidas para recomendação é a filtragem colaborativa [1]. Ela consiste em uma abordagem centrada no comportamento do usuário, pois usuários similares tendem a possuir padrões de comportamento semelhantes quando estão avaliando algo. Dessa forma, existem duas abordagens encontradas na filtragem colaborativa, a baseada em vizinhança de usuários, e a baseada em vizinhança de itens. Na abordagem baseada no usuário, as avaliações fornecidas por usuários similares são usadas para que sejam feitas recomendações para um usuário alvo. Por sua vez, a baseada em item consiste em encontrar um conjunto de itens similares a um item alvo, de modo que a predição desse item seja baseada em opiniões do usuário em relação aos outros itens do conjunto. Dessa maneira, utilizando a média dos pesos das avaliações, são computadas as supostas avaliações de um usuário para o item alvo.

Ambas abordagens de filtragem colaborativa estão inclusas no escopo de comparação com itens/usuários vizinhos. Esse tipo de abordagem possui alguns problemas, como falta de escalabilidade e partida fria. No primeiro caso, calcular a vizinhança entre itens ou entre usuários em bases de dados grandes pode ser um desafio, principalmente no caso da filtragem colaborativa baseada em usuários, cujas preferências e comportamento são mais dinâmicos e exigem um recálculo de similaridade com maior frequência [8]. A partida fria, por sua vez, é quando um novo usuário ou novo item é cadastrado no sistema, e o algoritmo não consegue calcular sua similaridade devido à falta de interações passadas.

Ambos os problemas de escalabilidade e partida fria estão presentes no cenário de recomendação baseada em sessão [7, 9]. Neste tipo de recomendação, os usuários não são identificados pelo sistema, sendo que são armazenadas apenas as sessões anteriores, que podem ter sido criadas pelos mesmos usuários ou por outros indivíduos. Assim, dada uma sessão, o sistema deve inferir as preferências do usuário com base apenas nas interações coletadas naquele momento. Neste caso, geralmente são utilizados dados implícitos, pois são mais abundantes e podem ser usados para identificar padrões de interesse. Por outro lado, dada a enorme quantidade de sessões

In: XVI Workshop de Trabalhos de Iniciação Científica (WTIC 2019), Rio de Janeiro, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2019.
ISSN 2596-1683

no sistema, a escalabilidade é afetada, principalmente quando utilizados algoritmos que calculam a vizinhança entre sessões.

Uma possível alternativa para reduzir ambos os problemas seria considerar informações de contexto da sessão a fim de realizar uma pré-amostragem dos itens que serão usados no cálculo da similaridade. Informação de contexto é qualquer informação que pode ser utilizada para descrever uma entidade do sistema, como usuário, item ou interação [3]. Como exemplo, o dia da semana, datas comemorativas, horário de acesso (dia, tarde ou noite), etc., são informações de contexto que podem melhorar a qualidade dos itens recomendados.

Desse modo, o presente artigo tem como finalidade desenvolver uma técnica de amostragem de itens e sessões baseada no contexto da sessão atual do usuário, de modo que o cálculo da similaridade possa ser realizado com uma quantidade menor de itens, e ao mesmo tempo, mais relacionados com as preferências do usuário daquela sessão.

As demais seções deste artigo estão organizadas como segue: na seção 2 são apresentados alguns conceitos relacionados com esta proposta; na seção 3 a proposta do artigo é descrita em detalhes; na seção 4 a metodologia de avaliação do sistema é apresentada; na seção 5 descrevem-se os resultados parciais obtidos até o momento; e por fim, na seção 6 as conclusões são expostas.

2 CONCEITOS RELACIONADOS

Nesta seção, são apresentados alguns conceitos relacionados com a proposta deste artigo, como informações de contexto e modelos de vizinhança.

2.1 Contexto na recomendação

Quando uma sessão de algum usuário é analisada, é possível extrair diversas informações extras. Isso se deve a utilização de contexto como um novo formato complementar no desenvolvimento de sistemas de recomendação, formato esse, que visa levar em consideração a interação que um ser humano tem com um sistema de recomendação. O conceito de contexto possui diversas interpretações e usos, segundo Dey, contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma entidade. Em uma utilização em sistemas de recomendação, uma entidade pode ser um usuário, um item ou uma experiência que um usuário esteja avaliando [11]. Como exemplo, assume-se o tempo como contexto, que, quando analisado, pode ser utilizado para a extração de informações como data da sessão, segundos, minutos, horas e dias. A partir da obtenção dessas informações, é possível identificar alguns padrões, e com isso, ajudar a recomendar novos itens baseados em situações semelhantes que outros usuários obtiveram acesso. Como exemplo, num contexto de e-commerce, identificou-se que no período da noite foram obtidos mais acessos a um item relacionado a televisão, portanto, quando algum usuário iniciar uma sessão, é mais provável que sejam recomendados itens relacionados ao identificado no padrão oriundo do contexto extraído para a recomendação. Vale ressaltar que um recomendador é nada mais que uma aplicação de estatística. O comportamento humano é confuso, e o principal desafio é descobrir padrões úteis entre a confusão que os dados podem oferecer [2]. Neste pensamento, surge a utilização dos contextos que cada sessão de um usuário oferece de modo a

melhorar a pré-amostragem dos itens que serão usados no cálculo da similaridade.

2.2 Modelos de vizinhança em SBRS

Em [10], os autores desenvolveram diversos modelos de vizinhança cujas performances foram relevantes frente aos *baselines* utilizados. O modelo *Session-Based KNN* (S-KNN), proposto pelos autores, estende o tradicional *Item-KNN* da seguinte maneira: para dada uma sessão s , um conjunto k de vizinhos mais próximos N_s , e uma função de similaridade $sim(s1, s2)$ que retorna um valor que determina a similaridade entre duas sessões $s1$ e $s2$, para que seja calculado o score de um item recomendável i , sendo este, representado por

$$score_{KNN}(i, s) = \sum_{n \in N} sim(s, n) \times 1_n(i)$$

em que $1_n = 1$ caso a sessão n possua o item i , e 0 caso contrário.

2.3 Escalabilidade do modelo S-KNN

Como já apontado pelos autores, os modelos de vizinhança apresentam problemas de escalabilidade, pois buscar os vizinhos mais próximos torna-se cada vez mais custoso à medida que o número de sessões aumenta, inviabilizando assim, a etapa de recomendação em cenários reais a longo prazo. Para contornar este problema, os autores utilizam estruturas de dados auxiliares que permitem recuperar as sessões que contém pelo menos um item em comum com a sessão atual, criando, assim, um conjunto de possíveis candidatos mais próximos. Como forma de reduzir ainda mais o espaço de busca, os autores propuseram duas estratégias de amostragem: recuperação de sessões mais recentes em relação a sessão atual e seleção de m sessões aleatórias. Foi verificado que, no cenário de *e-commerce*, as amostras com sessões mais recentes obtiveram uma considerável melhora na performance do algoritmo, o que indicaria que os algoritmos eram capazes de capturar tendências na comunidade de usuários. Contudo, como apontado em [5], o processo de ordenação das sessões mais recentes, também é uma variável de escalabilidade a ser considerada. Sendo assim, este projeto propõe alternativas, descritas na próxima sessão, para o uso de diferentes contextos a fim de amenizar esses problemas de escalabilidades.

3 ABORDAGEM PROPOSTA

Timestamps das interações são dados facilmente capturados e amplamente utilizado por SBRS para determinar ordem de acesso dos itens, o que permite a utilização de algoritmos baseados em sequência, tais como cadeias de Markov [10], ou então, como já mencionado na seção anterior, para amenizar problemas de escalabilidade já apontados em 2.3. Porém, a partir dos *timestamps* existentes na base de dados, também é possível extrair informações contextuais sobre a sessão atual de um usuário, tais como o dia da semana e a hora em que a interação ocorreu, entretanto, esse tipo de informação, ainda que facilmente obtida, não é explorada na literatura. Desta forma, este projeto propõe explorar a utilização de informações contextuais aplicadas como métodos de pré-filtragem, utilizando como *baseline* o modelo S-KNN, visto os problemas de escalabilidade já considerados na literatura. Além de informações de dia da semana e hora do acesso, foi considerado como contexto, o período do dia, dividido em 4 períodos:

- Manhã, das 6:00AM às 12:59PM;

- Tarde, das 1:00PM às 6:59PM;
- Noite, das 7:00PM às 11:59PM;
- Madrugada, das 0:00AM às 5:59AM.

4 METODOLOGIA

Uma das tarefas de um SRBS é fazer recomendações dos próximos itens a serem acessados durante a sessão atual de um usuário [5]. Sendo assim, verificamos a eficácia do uso de contextos temporais para a amostragem de sessões utilizando o mesmo protocolo de avaliação utilizado por Hidasi et al., que faz uma avaliação offline onde o recomendador produz uma lista de itens a serem recomendados a uma sessão, onde os primeiros n itens são revelados ao recomendador e o item seguinte é utilizado como item ideal da recomendação. Medimos a qualidade da lista produzida pelo recomendador a partir dos vinte itens mais relevantes para a sessão corrente, utilizando como métricas, a taxa de acerto (*HitRate*) e também pelo *Mean Reciprocal Ranking* (MRR), cujo cálculo é dado por:

$$MRR = \frac{1}{|LR|} \sum_{i=1}^{|LR|} \frac{1}{rank_i}$$

sendo LR a lista de itens relevantes, i um contador, e $rank_i$ a posição que o próximo item da sessão se encontra em LR.

4.1 Descrição dos Datasets

Os experimentos foram realizados nos datasets RSC15¹ e Retail-Rocket², que são dois datasets do domínio de *e-commerce* em que as empresas fazem recomendações como serviço e disponibilizaram publicamente *logs* de meses de interações dos usuários. Foram filtradas as sessões com apenas uma interação, assim como itens com menos de 5 acessos, assim como feito por Hidasi et al., porém, ao invés de separar apenas as sessões do último dia de interações do dataset para teste como os autores fizeram, utilizamos a última semana inteira de interações, a fim de poder analisar os contextos temporais envolvendo dias da semana. Dados gerais dos datasets estão exibidos na tabela 1.

Os datasets possuem grandes diferenças em relação aos contextos propostos em 3. A tabela 2 mostra que os usuários da Retailrocket tiveram menos acessos nos fins de semana, enquanto que a maior parte dos acessos se concentra no domingo e na segunda-feira no dataset RSC15.

Dados	Retailrocket	RSC15
Número de eventos	1.460.092	31.713.447
Número de sessões	366.425	7.986.524
Número de itens	60.627	37.486
Média de eventos por sessão	3,98	3,97

Tabela 1: Dados gerais dos datasets utilizados

No contexto de hora do dia, as figuras 1 e 2 indicam que provavelmente as plataformas possuem grupos de usuários bastante distintos, pois os horários de maior acesso são opostos. Enquanto que os horários de pico da Retailrocket se encontram durante a

¹<http://2015.recsyschallenge.com/challenge.html>

²<https://www.kaggle.com/retailrocket/ecommerce-dataset>

Dia	Retailrocket	RSC15
Seg	235.253	6.333.201
Ter	238.472	2.111.078
Qua	229.370	4.607.057
Qui	222.157	4.313.446
Sex	199.497	3.460.407
Sab	159.552	3.763.651
Dom	175.791	7.124.607

Tabela 2: Número de acessos por dia da semana

noite e madrugada, os horários do dataset RSC15 mostram picos de atividade durante a manhã e o fim da tarde.

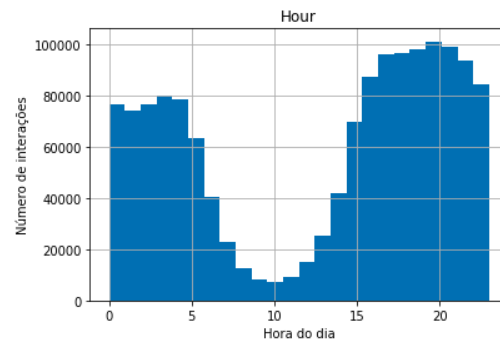


Figura 1: Número de Interações por hora no dataset Retail-Rocket)

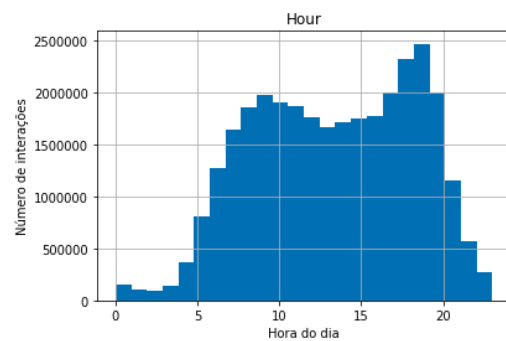


Figura 2: Número de Interações por hora no dataset RSC15

4.2 Ferramentas

Utilizamos a versão 3.5 da linguagem Python, juntamente com as bibliotecas NumPy e Pandas para o desenvolvimento deste projeto. Além disso, utilizamos como base a implementação do S-KNN

Período	Retailrocket	RSC15
Madrugada	450.379	1.673.042
Manhã	115.707	12.309.405
Tarde	417.153	11.249.254
Noite	476.853	6.481.746

Tabela 3: Número de Interações por período do dia.

disponibilizada por Ludewig and Jannach³, assim como suas implementações das métricas, para avaliar os resultados obtidos até então.

5 RESULTADOS PARCIAIS

Os resultados observados nas tabelas 4 e 5 foram obtidos utilizando o valor de $k = 500$, tamanho máximo da amostragem de 1000 sessões mais recentes, reportados em [10] como parâmetros que obtiveram melhores resultados.

Algoritmos	HitRate	MRR
Original	0.5957	0.3423
Dia da semana	0.4163	0.2524
Hora do dia	0.3031	0.1914
Período do dia	0.4922	0.2954

Tabela 4: Resultados dos contextos explorados no dataset da Retailrocket.

Algoritmos	HitRate	MRR
Original	0.6214	0.2758
Dia da semana	0.5726	0.2615
Hora do dia	0.5800	0.2644
Período do dia	0.6048	0.2710

Tabela 5: Resultados dos contextos explorados no dataset da RSC15.

A partir das tabelas 4 e 5, é possível observar que a utilização de contextos mais simples, como dia da semana e hora da interação, pioram o desempenho do modelo S - KNN original, porém, a queda de desempenho foi significativamente menor na base RSC15, onde o volume de dados de interações é muito maior em relação ao dataset Retailrocket. Contudo, os resultados contrastantes nas duas bases em relação aos contextos mencionados indicam que o volume de acessos, bastante distintos entre as bases, formam padrões de consumo. Além disso, a melhora dos resultados do contexto do período do dia em relação aos dois primeiros contextos indicam que, mesmo que a performance da amostragem por período do dia tenha sido inferior ao modelo original, contextos mais elaborados podem contribuir para uma amostragem melhor.

³<https://www.dropbox.com/sh/dbzmtq4zhbj5o9/AACldzQWbw-igKjcPTBI6ZPAa?dl=0>

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho propõe utilizar contextos temporais para amenizar os problemas de escalabilidade conhecidos em modelos de vizinhança quando aplicados em recomendações baseadas em sessões. Nossos experimentos iniciais indicam que estratégias simples de pré-filtragem como utilizar janelas de dia da semana ou horário do dia, comprometem a performance desses modelos, porém, a utilização de contextos mais elaborados resulta em selecionar melhores possíveis candidatos a sessões vizinhas mais próximas. Portanto, os próximos passos serão trabalhar com janelas de contexto mais elaboradas, como divisões por horário de trabalho, descanso e sono, ou encontrar padrões de acesso em datas comemorativas e feriados, por exemplo. Além disso, podemos aplicar contextos temporais em abordagens de pós-filtragem, fazendo estudos comparativos de performances entre abordagens de pré-filtragem e pós-filtragem em SRBS.

ACKNOWLEDGMENTS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, e dos recursos computacionais utilizados do supercomputador Euler, mantido pelo CEPID-CeMEAL.

REFERÊNCIAS

- [1] Charu C. Aggarwal. 2016. *Recommender Systems: The Textbook* (1st ed.). Springer Publishing Company, Incorporated.
- [2] Greg Linden Brent Smith. 2017. Two Decades of Recommender Systems at Amazon.com. *IEEE INTERNET COMPUTING* (2017).
- [3] Anind K. Dey. 2001. Understanding and Using Context. *Personal Ubiquitous Comput.* 5, 1 (Jan. 2001), 4–7. <https://doi.org/10.1007/s007790170019>
- [4] Anind K. Dey. 2001. Understanding and Using Context. *Personal and Ubiquitous Computing* (2001).
- [5] Hui Feng Guo, Ruiming Tang, Yunming Ye, Feng Liu, and Yuzhou Zhang. 2019. A Novel KNN Approach for Session-Based Recommendation. In *Advances in Knowledge Discovery and Data Mining*, Qiang Yang, Zhi-Hua Zhou, Zhiguo Gong, Min-Ling Zhang, and Sheng-Jun Huang (Eds.). Springer International Publishing, Cham, 381–393.
- [6] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [7] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks Meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*. ACM, New York, NY, USA, 306–310. <https://doi.org/10.1145/3109859.3109872>
- [8] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender Systems: An Introduction* (1st ed.). Cambridge University Press, New York, NY, USA.
- [9] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of Session-based Recommendation Algorithms. *arXiv preprint arXiv:1803.09587* (2018).
- [10] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction* 28, 4 (01 Dec 2018), 331–390. <https://doi.org/10.1007/s11257-018-9209-6>
- [11] Pedro G. Campos Soto. 2013. Recommender Systems and Time Context: Characterization of a Robust Evaluation Protocol to Increase Reliability of Measured Improvements. (2013).