

UM ALGORITMO DE RECOMENDAÇÃO UTILIZANDO GRAFOS¹

Leonardo Dias²

José Antônio de Figueiredo³

RESUMO

Qualquer tipo de usuário na Internet possui itens que são de seu interesse, porém nem sempre eles encontram esse tipo de informações, normalmente porque não possuem conhecimento de que tal informação está disponível devido ao grande volume existente. Sistemas de recomendação buscam resolver esse problema. Com isso em mente foi desenvolvido um algoritmo de recomendação utilizando a teoria dos grafos e utilizando a similaridade entre cossenos para verificar a similaridade entre os conjuntos obtidos. Os resultados obtidos podem ser considerados como um sucesso, pois as recomendações foram realizadas de maneira correta.

Palavras-chave: teoria dos grafos, sistemas de recomendação, javascript, nodejs

1 INTRODUÇÃO

A Internet atualmente contém diversos tipos de websites e produtos com finalidades diferente, sejam elas conectar pessoas, por meio das redes sociais, ou sejam elas vender produtos ou apenas trazer informação. Os sistemas de recomendação podem auxiliar em qualquer que seja a finalidade desejada por essas plataformas, sempre melhorando a mesma. No caso das redes sociais, podem ser sugeridas pessoas que possuem conhecidos em comum, em caso de e-commerces, podem ser sugeridos itens que os usuários possam ter interesse.

Este trabalho visa desenvolver um desses sistemas de recomendação utilizando a Teoria dos Grafos, uma teoria da matemática que estuda as relações entre os objetos de um conjunto. Para isso foi desenvolvido um sistema de

¹ Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet, na cidade de Passo Fundo, em 2018.

² Graduando em Tecnologia de Sistemas para Internet pelo IFSUL campus Passo Fundo. Email: leo.lbdias@gmail.com

³ Orientador, professor do IFSUL. Email: jose.figueiredo@passofundo.ifsul.edu.br

recomendação genérico, permitindo com que qualquer tipo de dado seja utilizado, sendo necessário apenas um conjunto de usuários com seus interesses já salvos.

As demais partes deste artigo estão dispostas da seguinte forma: a Seção dois aborda o desenvolvimento do trabalho, iniciando pela explicação do que é um sistema de recomendação, assim como o tipo de filtragem que foi utilizada, seguido por uma explicação do que é a teoria dos grafos. Para finalizar a seção descreve o que é a comparação entre cossenos e o estudo da tecnologia utilizada. A segunda seção descreve a criação do projeto, englobando a criação do diagrama de classes e a realização dos testes. Por fim, na terceira seção estão as considerações finais e os trabalhos futuros.

1 REFERENCIAL TEÓRICO

O desenvolvimento do sistema começou com o levantamento de referências sobre termos abordados no projeto, como a teoria dos grafos, seguido pelo estudo de tecnologias. Por fim, foi feita a conceptualização, a modelagem e a codificação do algoritmo.

2.1 SISTEMAS DE RECOMENDAÇÃO

Sistemas de recomendação são programas que procuram recomendar os itens mais adequados para as necessidades de usuários particulares, prevendo os interesses dele a partir de informações existentes que foram resultadas de interações entre os usuários e os itens. LU, et al (2015).

De acordo com Aggarwal(2016), o objetivo primário de um sistema de recomendação é o lucro, pois a partir deles é possível recomendar itens que os usuários estão interessados até eles. Porém, nessa plataforma, isso será utilizado para melhorar a experiência do usuário.

Partindo disso, os algoritmos de recomendação são divididos em três categorias: Filtragem Colaborativa, Filtragem Baseada em Conteúdo e Filtragem

Híbrida. Apenas a Filtragem Colaborativa será explicada, sendo essa utilizada para realizar esse trabalho.

2.1.1 FILTRAGEM COLABORATIVA

A ideia desse tipo de filtragem é que se alguma pessoa 'A' tem os mesmos interesses de outra pessoa 'B', sendo que isso é calculado através de compras realizadas ou visualizações em itens, por exemplo, eles também terão esses mesmos interesses no futuro. Isso significa que se a pessoa 'B' comprar um livro, este mesmo livro seria provavelmente indicado a pessoa 'A' caso ela ainda não tenha visualizado este. JANNACH, et al (2010, p. 20-21).

Zieseimer (2012, p.33) diz que os sistemas de filtragem colaborativa são divididos em algoritmos baseados em memória, que apenas utilizam de técnicas para identificar usuários com comportamentos similares, e os baseados em modelos, que utilizam de coleções de avaliações para aprender um modelo de perfil para apenas depois receber as recomendações de conteúdo.

2.1.2 ALGORITMOS BASEADOS EM MODELOS

Adomavicius e Tuzhilin (2005, p. 737) explicam que os algoritmos baseados em modelo são desenvolvidos utilizando *machine learning* e técnicas de estatísticas baseados em um modelo de dados.

Nesse tipo de abordagem o sistema precisa aprender sobre o modelo do perfil do usuário, coisa que os algoritmos baseados em memória não fazem, para poder recomendar o conteúdo que interessa a este usuário.

2.1.3 ALGORITMOS BASEADOS EM MEMÓRIA

Neste tipo de algoritmo, é levado em consideração as similaridades entre os usuários para realizar uma recomendação baseada em seus interesses em comum. (ZIESEMER, 2012, p.33). Isso acaba sendo interessante pois não é necessário traçar um perfil para cada um dos usuários do sistema.

Isso significa que o sistema irá procurar usuários semelhantes e recomendar o que um acaba priorizando para o outro. Ou, também, o sistema pode procurar um usuário que gosta exatamente do oposto de algum outro usuário, para então poder traçar o que ele não deve recomendar a ele. O algoritmo criado no trabalho utilizou desta técnica.

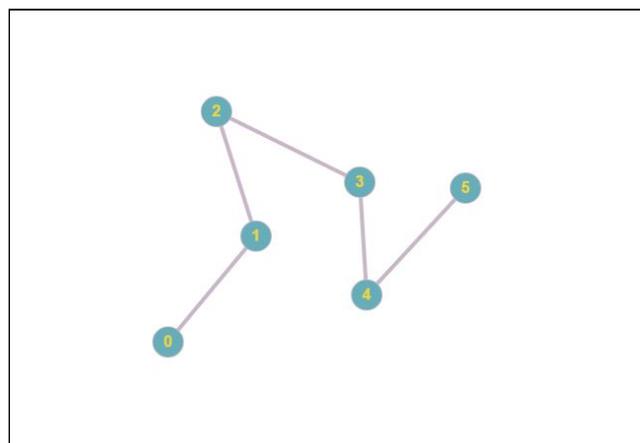
2.2 TEORIA DOS GRAFOS

A teoria dos grafos é apenas um conceito da matemática que estuda as relações entre os objetos de um conjunto qualquer. Feofiloff (2011, p. 08) explica que um grafo é um par (V, A) , que também pode ser representado por $G(V, A)$, onde V , os vértices, são um conjunto arbitrário e A , as arestas, é um subconjunto de $V^{(2)}$. Existem diversos tipos de grafos, porém os principais são:

2.2.1 GRAFO SIMPLES

Feofiloff(2011) diz que um grafo simples é um grafo não direcionado, sem laços, onde existe apenas uma aresta entre cada vértice. A Figura 1 demonstra um desenho de como um grafo simples é representado.

Figura 1 – Um grafo simples

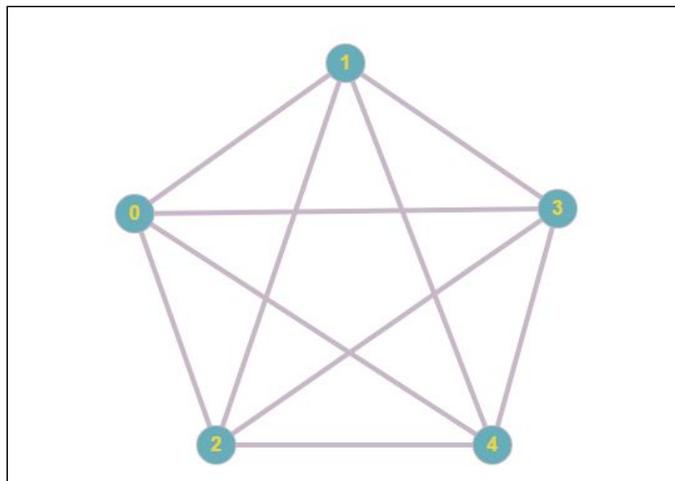


Fonte: Do Autor

2.2.2 GRAFO COMPLETO

É um grafo simples em que todos os vértices estão conexos a todos os outros vértices. Feofiloff (2011, p. 09) define um grafo completo como “Um grafo G é completo se $A(G) = V(G)^{(2)}$ ”. A Figura 2 demonstra a representação gráfica de um grafo completo.

Figura 2 – Um grafo completo



Fonte: Do Autor

2.3 SIMILARIDADE ENTRE COSSENO

A neo4j (2018) define a similaridade entre cossenos como o ângulo entre dois vetores n -dimensionais em um espaço n -dimensional. A fórmula seria o produto escalar dos dois vetores dividido pelo produto do tamanho destes. A Figura 3 representa graficamente a fórmula.

Figura 3 – Fórmula da similaridade entre cossenos

$$\text{similarity}(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

Fonte: Site da neo4j

3 MATERIAIS E MÉTODOS

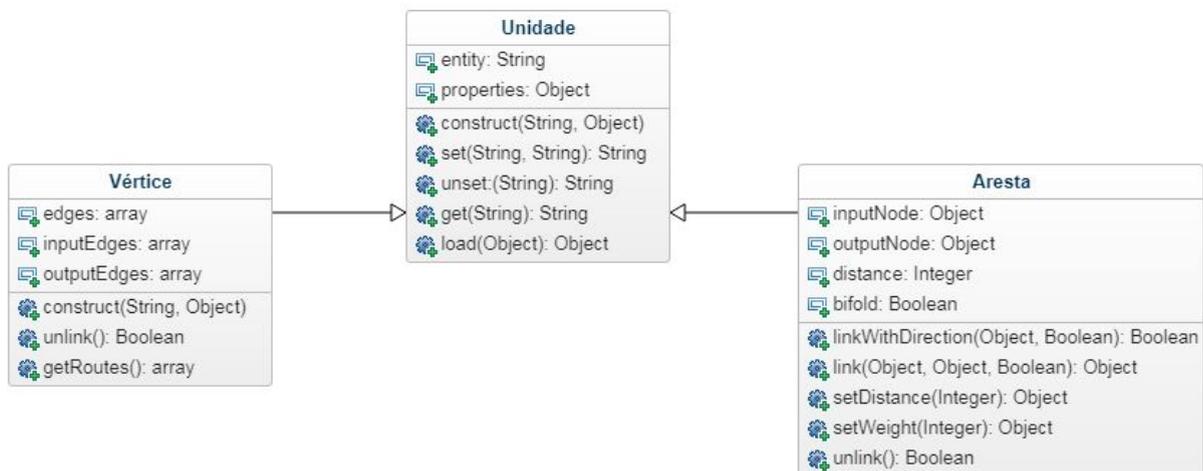
A pesquisa realizada pode ser classificada como exploratória, uma vez que buscou a familiarização com áreas como a Teoria dos Grafos e os algoritmos de recomendação. A aplicação da teoria dos grafos na construção do algoritmo se deu com a utilização de classes da própria linguagem JavaScript, assim como a utilização de algoritmos, como a similaridade dos cossenos para poder obter um coeficiente, assim podendo realizar as recomendações.

A pesquisa foi conduzida, em duas etapas. Na primeira foram buscados em materiais como artigos e tutoriais, onde foram estudados os conceitos referente a teoria dos grafos, sistemas de recomendação. A segunda etapa englobou principalmente os estudos da linguagem utilizada e a aplicação destas no desenvolvimento dos grafo e obtenção das recomendações.

3.1 MODELAGEM DO GRAFO

O desenvolvimento do algoritmo foi dividido em duas partes, onde a primeira se focou em conseguir realizar a criação dos grafos em memória. Isso foi criado utilizando as classes do JavaScript, assim como seus métodos. O diagrama de classes da Figura 4 apresenta a modelagem do algoritmo.

Figura 4 – Diagrama de classes



Fonte: Do Autor

A classe Unidade foi criada apenas para que seja possível identificar os vértices e arestas através da *entity* e *properties*. A *entity* apenas define o tipo, onde no caso do trabalho, os vértices podem ser pessoas ou jogos e as arestas apenas são representadas por *like*. Os métodos estão presentes apenas para carregar, remover e verificar as propriedades.

Os vértices possuem as arestas de saída e entrada, respectivamente representadas através dos atributos *outputEdges* e *inputEdges*, o método *unlink* apenas remove as ligações que o vértice possui e o *getRoutes* será explicado mais adiante.

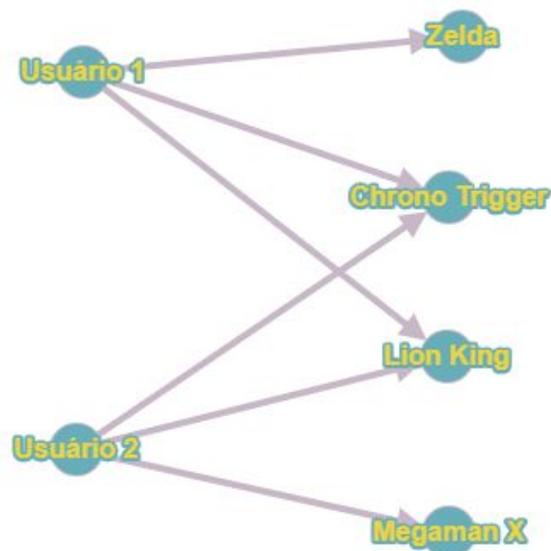
A classe Aresta é onde acontece toda a lógica do algoritmo, que é feita através do método *link*. Ele recebe três parâmetros, sendo que dois deles são vértices e o terceiro apenas representa se a ligação é ambidirecional ou não. O

primeiro parâmetro é o vértice de entrada e o segundo, o de saída, ambos são respectivamente representados através dos atributos *inputNode* e *outputNode*. O terceiro parâmetro é representado pelo atributo *bifold*. A direção é definida através do método *linkWithDirection*, que é chamado pelo atributo *link*. Também é possível definir a distância e o peso da ligação através dos métodos *setDistance* e *setWeight*.

4 RESULTADOS

O que realiza a maior parte das operações no algoritmo é o método *link* na classe Aresta. O método realiza a ligação de um vértice com outro, já informando a direção da ligação através do método *linkWithDirection*, onde o primeiro parâmetro é o vértice de entrada, que é chamado de *inputNode*, e o segundo é o de saída, o *outputNode*. A imagem 5 representa o resultado da ligação de diversos vértices através de arestas, onde as mesmas representam um *like*, resultando na criação de um grafo. O grafo representa dois usuários que gostam de diversos jogos.

Figura 5 – Representação de ligações de vértices com outros através de arestas



Fonte – Do autor

4.1 OBTENDO RECOMENDAÇÕES Para executar as recomendações, primeiro é necessário recuperar todas as iterações que um vértice possui através do método *getRoutes*. Esse método retorna um conjunto com todos os nomes dos vértices

relacionados ao que chamou o mesmo, isso é representado pelo log apresentado pela Figura 6.

Figura 6 – O resultado do método *getRoutes* de dois usuários diferentes

```
[ 'Lion King', 'Chrono Trigger', 'The Legend of Zelda' ]
//
[ 'Lion King', 'Chrono Trigger', 'Megaman X' ]
```

Fonte – Do autor

A partir disso, é possível realizar a comparação entre os conjuntos que se obtém através desse método utilizando a similaridade entre cossenos. Esse algoritmo retorna um coeficiente de 0 à 1, onde 0 representa que os vértices não possuem nada em comum, e 1 representa que elas são idênticas. O resultado da similaridade dos dois conjuntos acima é 0.6666666667

Com esse resultado da similaridade entre os usuários é possível verificar se eles são parecidos o suficiente para realizar a recomendação. Como o método utilizado foi a filtragem colaborativa baseada em memória, caso os usuários sejam parecidos o suficiente, é necessário realizar a diferença entre os conjuntos dos usuários, onde o resultado é mostrado na Figura 7, para após realizar as recomendações.

Figura 7 – Resultado da diferença dos conjuntos

```
[ 'The Legend of Zelda', 'Megaman X' ]
```

Fonte – Do autor

O resultado final está sendo representado na Figura 8, onde é apenas o resultado de ambos usuários após a recomendação ser feita. A recomendação retorna um conjunto para cada usuário, onde os itens que não foram gostados por um usuário mas que foram gostados por seu semelhante. Como se pode ver, os dois conjuntos finais são exatamente iguais, pois a diferença entre os dois antes da recomendação ocorrer era de apenas um item em cada um dos conjuntos.

Figura 8 – Resultado das recomendações

```

Array final do primeiro usuário:
[ 'Lion King',
  'Chrono Trigger',
  'The Legend of Zelda',
  'Megaman X' ]
=====
Array final do segundo usuário:
[ 'Lion King',
  'Chrono Trigger',
  'Megaman X',
  'The Legend of Zelda' ]

```

Fonte – Do autor

5. TESTES

Para realizar os testes, foram criados quatro usuários e 20 itens, onde cada um é um vértice, assim como vinte arestas para realizar as ligações entre os mesmos. Para cada usuário foram inseridos cinco itens, onde esses foram recuperados randomicamente da coleção dos itens. Os conjuntos completos são representados pela Figura 9.

Figura 9 – Resultado dos itens que os usuários gostaram através da randomização

```

Array do primeiro Usuário
[ 'Star Fox',
  'Mario Kart',
  'Super Bomberman',
  'The Legend of Zelda' ]
====//====
Array do segundo Usuário
[ 'F-Zero',
  'Chrono Trigger',
  'Final Fantasy 5',
  'Kirby Super Star' ]
====//====
Array do terceiro Usuário
[ 'Donkey Kong Country',
  'Megaman X',
  'Super Bomberman',
  'Chrono Trigger' ]
====//====
Array do quarto Usuário
[ 'Star Fox',
  'The Legend of Zelda',
  'Donkey Kong Country',
  'Chrono Trigger' ]

```

Fonte – Do autor

Após, foram realizadas as comparações dos conjuntos utilizando a similaridade dos cossenos do primeiro usuário com o restante, e caso a mesma for maior ou igual à 0.49, a diferença entre os conjuntos será feita como anteriormente, assim como a recomendação. O valor de 0.49 foi escolhido pensando que os usuários deveriam possuir uma semelhança de pelo menos 50% para poder realizar a recomendação, porém qualquer outro valor poderia ser atribuído. Ambos são representados pela Figura 11.

Figura 11 – Resultado dos itens que os usuários gostaram através da randomização

```
Similaridade dos cossenos entre o primeiro e o segundo usuário
0
=====//=====
Similaridade dos cossenos entre o primeiro e o terceiro usuário
0.25
=====//=====
Similaridade dos cossenos entre o primeiro e o quarto usuário
0.5
Diferença entre o primeiro e o quarto usuário
[ 'Mario Kart',
  'Super Bomberman',
  'Donkey Kong Country',
  'Chrono Trigger' ]
Realizando as recomendações baseadas no quarto usuário
[ 'Star Fox',
  'Mario Kart',
  'Super Bomberman',
  'The Legend of Zelda',
  'Donkey Kong Country',
  'Chrono Trigger' ]
```

Fonte – Do autor

Como pode ser notado pelo log, apenas o quarto usuário é semelhante o suficiente ao primeiro para que as recomendações sejam feitas. Sendo assim, os itens “Chrono Trigger” e “Donkey Kong Country” foram adicionados ao conjunto, e caso fosse necessário recomendar ao segundo usuário, “Mario Kart” e “Super Bomberman” poderiam ser recomendados ao mesmo.

6. CONSIDERAÇÕES FINAIS

O objetivo principal do trabalho sempre esteve voltado ao desenvolvimento de um algoritmo de recomendação utilizando grafos e a técnica de filtragem colaborativa. Sendo assim, pode-se considerar que os resultados foram positivos.

Mesmo que existam outras maneiras de realizar recomendações, a abordagem utilizando grafos pode ser de grande utilidade para quando se necessita recomendar itens de uma base de dados menor, pode-se utilizar de exemplo as bases de *start-ups*, pois não é necessário realizar todas as operações diretamente no banco de dados. No caso da abordagem estudada no trabalho seria necessário apenas buscar os dados de cada tabela para realizar a criação os grafos, realizando o restante em memória, resultando em uma melhoria na performance.

Entre as principais melhorias que podem motivar trabalhos futuros, seria de extrema relevância desenvolver um método para que as recomendações sejam feitas automaticamente, sem que seja necessário informar quais usuários devem ser comparados manualmente. Para isso seria necessário estudar métodos para percorrer grafos, como o algoritmo de Dijkstra's. Também seria importante salvar o grafo em disco, para não precisar realizar a criação do mesmo sempre que for necessário realizar recomendações.

ABSTRACT

Any user on the internet has interest on any type of item, but not always they can find that information, normally because they don't have knowledge that the information is available due to the big volume of information that exists. Recommendation systems seek to solve this problem. With this in mind, a recommender system using graph theory was developed, utilizing the cosine similarity to verify the similarity between arrays that were obtained. The results obtained could be considered successful, because the recommendations were made correctly.

Keywords: graph theory, recommender systems, javascript, nodejs

REFERÊNCIAS

ADOMAVICIUS, G. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. Disponível em: <<http://blog.ag-nbi.de/wp-content/uploads/2015/10/adomavicius-recsys.pdf>>. Acesso em 8 Out. 2017.

AGGARWAL, C.C, Recommender Systems: The Textbook, Springer International Publishing: 2016.

DAYLEY, Brad. Node.js, MongoDB and AngularJS: Web Development, Pearson Education, Ann Arbor: 2014.

GOOGLE DEVELOPERS, Introduction to Chrome V8, Disponível em: <<https://developers.google.com/v8/intro/>>. Acesso em 14 Out. 2017.

JANNACH, Dietmar, ZANKER, Markus, FELFERNIG, Alexander. Recommender Systems: An Introduction. Cambridge University Press: 2010 por Dietmar Jannach (Autor), Markus Zanker (Autor), Alexander Felfernig (Autor)

LU, Jie, et al. Recommender System Application Developments: A Survey. 2015.

Disponível em:

<<https://www.uts.edu.au/sites/default/files/desi-publication-recommender%20system%20application%20developments%20a%20survey-accepted%20manuscript.pdf>>

Acesso em: 8 Out. 2017

NEO4J, The Cosine Similarity algorithm, Disponível em:

<<https://neo4j.com/docs/graph-algorithms/current/algorithms/similarity-cosine/>>.

Acesso em 10 Nov. 2018.

P. Feofiloff, Y. Kohayakawa, Y. Wakabayashi, Uma Introdução Sucinta à Teoria dos Grafos, Disponível em:

<<https://www.ime.usp.br/~pf/teoriadosgrafos/texto/TeoriaDosGrafos.pdf>>. Acesso

em: 27 Out. 2018