Uma abordagem user-based do algoritmo K-nearest Neighbours

Lívia de Azevedo[1], **Gustavo Ebbo**[1], **Ivo S. Paiva** [1]

[1]Departamento de Ciência da Computação – Universidade Federal Rural do Rio de Janeiro (UFRRJ) – Nova Iguaçu – RJ – Brasil

{livia_de_azevedo,gustavoebbo,ivopaiva}@ufrrj.br

Abstract. *Among the many approaches of recommender systems, the collaborative filtering ones have aroused a lot of interest due to its easy implementation, as well as the possibility of transpose it to a real world environment. In this article we approach a KNN implementation, one of the main algorithms in the literature, and evaluate four similarity methodologies amoung our test base,presenting each one's efficiency.*

Resumo. *Dentre as inúmeras abordagens de Sistemas de Recomendação, as de Filtragem Colaborativa têm despertado bastante interesse, devido à sua facilidade de implementação, assim como a possibilidade de transpô-la para uma situação do mundo real. Neste artigo, abordamos uma implementação do KNN, um dos principais algoritmos utilizados na literatura, e avaliamos quatro metodologias de similaridades dentro de nosso conjunto testes, apresentando a eficiência de cada um.*

1. Introdução

Com o advento das lojas *online*, a conexão entre os lojistas e o consumidor final apresentou uma considerável melhora, uma vez que o usuário pode comparar preços em lojas distintas em minutos, inclusive em outros países, além de possibilitar ao lojista um estoque "infinito".

O benefício, no entanto, pode tornar-se um incomodo, pois de acordo com a pesquisa [Schwartz 2004], quando uma pessoa possui à sua disposição muitas opções, esta tende à indecisão e não escolher nenhuma das opções. A partir da observação destas situações, surgiu a necessidade de oferecer ao consumidor um "recomendador", algo que avaliasse o perfil do consumidor e de acordo com determinados parâmetros, otimizasse a busca dos objetos de interesse e apresentasse resultados com maior probabilidade de aceitação e que, portanto, aumentasse a projeção das vendas para o lojista.

A área de Sistemas de Recomendação visa automatizar as funções de um recomendador, fazendo com que seja possível a recomendação em lojas online tal como observamos em lojas físicas. Para tanto, existem inúmeros métodos de obter os parâmetros utilizados para a recomendação assim como realizar a recomendação propriamente dita. Neste artigo abordaremos o método de filtragem colaborativa por meio do KNN (K-nearest Neighbours).

A filtragem colaborativa consiste em observar na base dados quais usuários mais se assemelham ao usuário para qual é desejado a recomendação, partindo do principio de

que se um usuário A apresenta gostos e interesses similares aos de um usuário B, então se o usuário A gostou de um item i, o usuário B também gostará do item i.

Nas próximas seções, serão abordados o algorítimo K-Nearest Neighbours, métricas de similaridades consideradas neste trabalho, métrica de avaliação, resultados e conclusão.

2. K-nearest Neighbours

O algoritmo do *K-nearest Neighbours*(K-vizinhos mais próximos) é pertencente a um grupo de algoritmos na Filtragem Colaborativa denominados como *Memory-based*. A ideia básica do algoritmo consiste na obtenção dos k vizinhos com características mais próximas a um determinado ponto(neste caso, a um usuário). Existem modelos de classificação, que observam a maior quantidade de vizinhos com um determinado rótulo e preveem o ponto como sendo pertencente a este grupo de vizinhos, e de regressão, que utilizam alguma estratégia para calcular um valor que define a intensidade de semelhança com base nos rótulos dos elementos já conhecidos e utilizam este valor para ranquear os elementos mais próximos ao ponto analisado [Desrosiers and Karypis 2011]. O modelo escolhido para a implementação do KNN neste relatório foi o de regressão, que ranqueia os vizinhos utilizando métricas de similaridades (que serão explicadas na próxima seção) para definir os k vizinhos mais próximos de um usuário.

Para prever uma nota $\hat{r}_{ui}$ dado um usuário u e um item i, foi utilizada a normalização denominada *Z-score* (representado pela função $h(r_{ui})$), com objetivo de converter as notas geradas por cada usuário para uma escala mais homogênea possibilitando uma comparação direta. Em outras palavras, cada usuário tem sua forma intrínseca de dar uma nota para um item e com isso uma escala que unifique estes comportamentos é necessária, para tal, foi utilizado o *Z-score*. A abordagem será da seguinte forma:

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \frac{\sum_{v \in N_i(u)} w_{uv} h(r_{ui})}{\sum_{v \in N_i(u)} |w_{uv}|}$$

onde:

$\bar{r}_u$: É a média das notas dos itens que foram avaliados pelo usuário u;
σ_u: É o desvio padrão das notas dos itens que foram avaliados pelo usuário u;
$N_i(u)$: É o conjunto de vizinhos mais próximos de u;
w_{uv}: É a similaridade entre os usuários u e v;
$h(r_{ui})$: *Z-score*, definido como: $h(r_{ui}) = \dfrac{r_{ui} - \bar{r}_u}{\sigma_u}$

2.1. Métricas de similaridade

Para este experimento, escolhemos quatro medidas de similaridades para avaliação: Cosseno, Correlação de Pearson, Distância Euclidiana e a Diferença Quadrática Média.

Em nossos testes, consideramos que as similaridades só serão calculadas se, entre os usuários u e v, houver pelo menos 10 itens em comum. O motivo disto foi para evitar cálculos com poucos usuários que poderiam piorar a previsão mas em compensação a cobertura do algoritmo é diminuída.

2.1.1. Cosseno

A medida de similaridade entre dois elementos a e b consiste em representá-los em dois vetores X_a e X_b e aplicar a fórmula abaixo:

$$cos(X_a, X_b) = \frac{X_a^T X_b}{\|X_a\| \, \|X_b\|}$$

Essa medida pode ser utilizada para calcular similaridades entre usuários representando um usuário u como um vetor $X_u \in \mathbb{R}^{|I|}$,tal que $x_{ui} = r_{ui}$ se o usuário u avaliou o item i, e 0 caso contrário. A similaridade entre dois usuários é calculada a partir da fórmula abaixo:

$$w_{ui} = cos(X_u, X_v) = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in I_u} r_{ui}^2 \sum_{j \in I_v} r_{vj}^2}}$$

Onde I_{uv} representa os itens avaliados por ambos usuários u e v. O problema desta similaridade é não considerar a média e variância das avaliações dos usuários.

2.1.2. Correlação Pearson

Uma medida bem conhecida que compara as notas onde os efeitos da média e da variância são removidos é a similaridade da Correlação de Pearson, dada pela fórmula abaixo:

$$w_{uv} = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}}$$

A correlação pode ser direta ou inversa e os valores variam entre 0 e 1.

2.1.3. Distância Euclidiana

A similaridade da distância euclidiana de um usuário u e v é dada como:

$$w'_{uv} = \sqrt{\sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2}$$

Logo em seguida, para normalizar o valor(colocar no range entre 0 e 1), aplicamos a seguinte transformação:

$$w_{uv} = \frac{1}{1 + w'_{uv}}$$

2.1.4. Diferença Quadrática Média

A similaridade da Diferença Quadrática Média entre um usuário u e v é dada como:

$$w'_{uv} = \frac{|I_{uv}|}{\sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2}$$

Para a normalização do valor:

$$w_{uv} = \frac{1}{1 + w'_{uv}}$$

3. Resultados

Todos os testes foram avaliados pela função *Mean Absolute Error*, que calcula a média absoluta dos erros das previsões feitas no conjunto de testes($test\_set$):

$$MAE = \frac{\sum_{u,i \in test\_set} |r_{ui} - \hat{r}_{ui}|}{|test\_set|}$$

Os testes foram executados em um computador com *Windows* 10 de 64 bits, 8GB de memória RAM, processador Intel Core i5-3230 de 2.60GHz e compilador do Julia versão 0.4.6. A lógica de divisão da base de dados seguiu o método do *K-Fold Cross Validation*(KFCV), sendo dividido em cinco vezes a base de dados para o KFCV, resultando em 80% da base para treinamento e 20% para teste. Além do MAE foram observados o valor da cobertura de cada execução, que corresponde a taxa média de alcance de todos os usuários do conjunto de teste, e o desvio padrão da quantidade de elementos do conjunto de testes que não foram previstos, durante uma execução do algoritmo. Para a construção dos resultados, foram realizados testes variando os valores de k em intervalos de de diferença de 10, começando de 10 até 50. Para cada um dos 4 casos que foram testados para cada similaridade, a ordem dos elementos da base de dados foram alternadas aleatoriamente, tornando os casos de testes imprevisíveis para a avaliação da acurácia de generalização do algoritmo. Foram calculados também a média de cada uma das medidas já citas anteriormente e organizadas nos gráficos 1, 2, 3 e nas tabelas 1,2,3,4 e 5.

Métricas	MAE	Cobertura	Desvio Padrão (Não Cobertos)
Cosseno	0.73668	94.54850	41.82836
Pearson	0.74449	94.47750	52.59729
Dist. Euclidiana	0.77560	94.55800	35.70747
Dif. Quad. Média	0.81943	94.48950	29.94187

Tabela 1. Valores aproximados obtidos para K=10

Métricas	MAE	Cobertura	Desvio Padrão (Não Cobertos)
Cosseno	0.72168	88.43325	47.03241
Pearson	0.72656	88.34775	48.83493
Dist. Euclidiana	0.75568	88.44400	72.84915
Dif. Quad. Média	0.78976	88.40425	54.20221

Tabela 2. Valores aproximados obtidos para K=20

Métricas	MAE	Cobertura	Desvio Padrão (Não Cobertos)
Cosseno	0.71701	82.29700	50.85684
Pearson	0.71916	82.10700	43.74030
Dist. Euclidiana	0.74774	82.35425	69.03393
Dif. Quad. Média	0.77742	82.36950	56.94406

Tabela 3. Valores aproximados obtidos para K=30

Métricas	MAE	Cobertura	Desvio Padrão (Não Cobertos)
Cosseno	0.71533	76.17050	71.18038
Pearson	0.71650	75.98475	48.96655
Dist. Euclidiana	0.74355	76.17000	46.10631
Dif. Quad. Média	0.76962	76.12075	52.67557

Tabela 4. Valores aproximados obtidos para K=40

Métricas	MAE	Cobertura	Desvio Padrão (Não Cobertos)
Cosseno	0.71312	70.08300	77.92084
Pearson	0.71356	69.93800	84.81883
Dist. Euclidiana	0.73960	70.13975	65.23899
Dif. Quad. Média	0.76373	70.16625	76.82323

Tabela 5. Valores aproximados obtidos para K=50

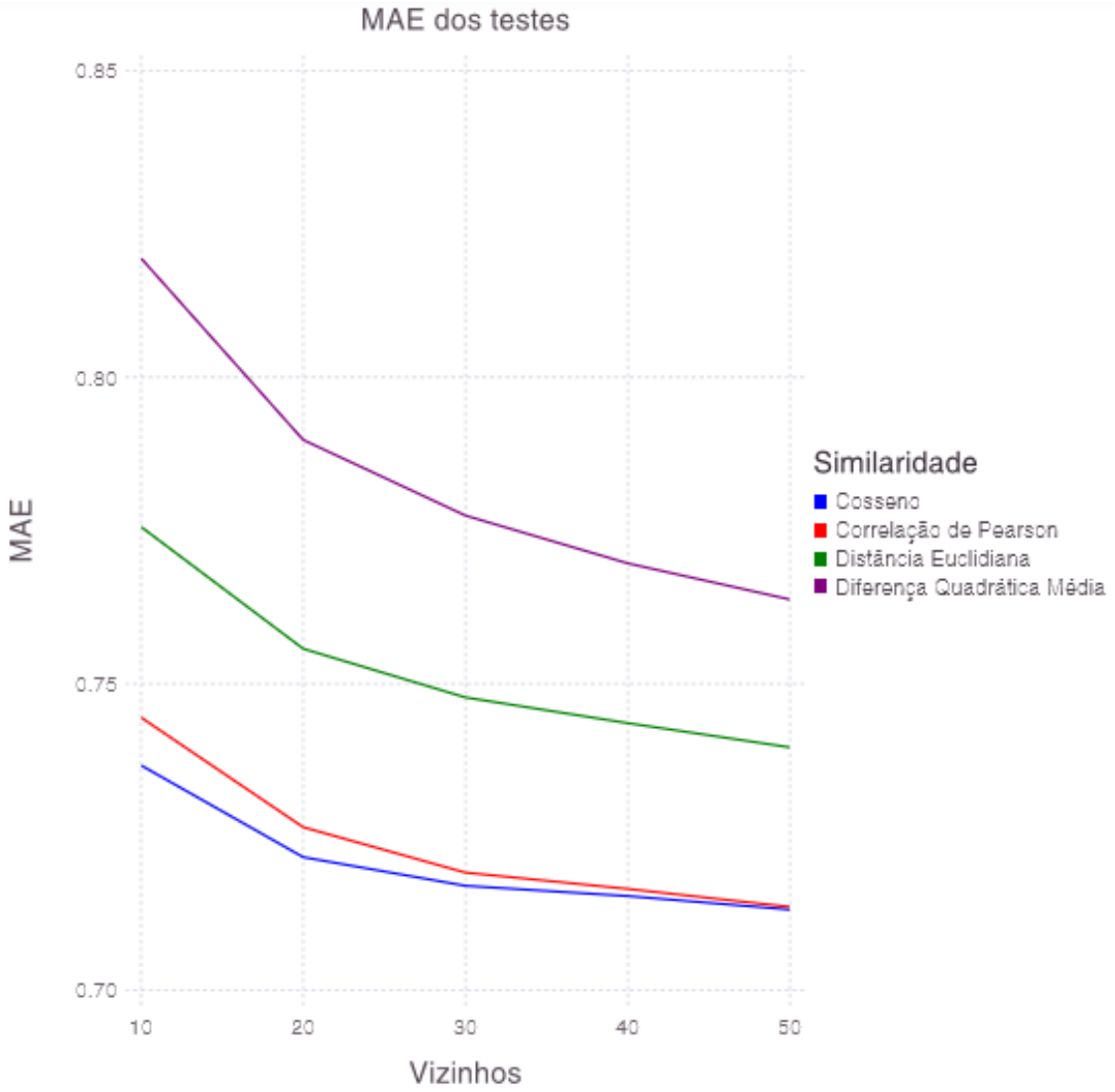

Figura 1. Gráfico das médias dos valores do MAE

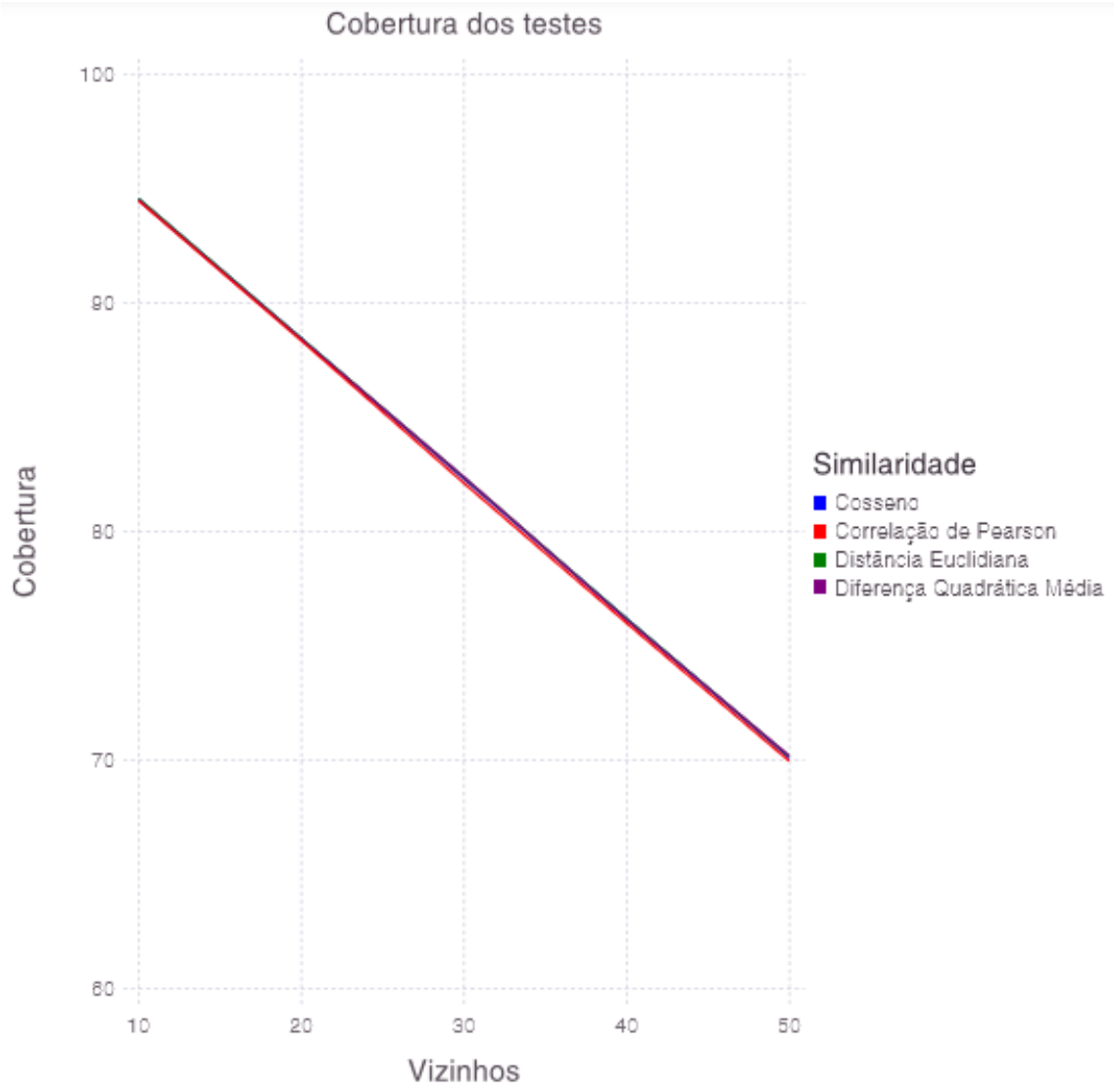

Figura 2. Gráfico das médias dos valores do Cobertura

4. Conclusão e trabalhos futuros

De acordo com os resultados obtidos, é possível notar que quanto maior a quantidade k de vizinhos, menor é a cobertura. Esse fenômeno acontece nesta base de dados de 100.000 avaliações, pois a base oferece uma pequena quantidades de usuários. Desta forma, uma quantidade grande de vizinhos diminuem a cobertura, pois não há, na maioria dos casos, a respectiva quantidade k de usuários semelhantes no conjunto de testes. Nessa base, também é possível notar que o desvio padrão de usuários não cobertos tende a aumentar com a maior quantidade de vizinhos, pois esses usuários não cobertos têm menor similaridade. Para diminuir o erro médio absoluto (MAE), todas as notas foram arredondadas, pois na base só existem notas inteiras. As similaridades de Cosseno e Pearson obtiveram os melhores resultados de MAE. Ambos obtiveram resultados similares no MAE quando o valor de k aumentava, porém o desvio padrão médio dos usuários não cobertos foi, na maioria dos testes, menores na similaridade do Cosseno.

Como trabalhos futuros, serão abordadas outras similaridades e uma avaliação

mais profunda nas diferenças de resultados obtidos em cada similaridade. Assim como aplicar os testes com outras métricas de avaliação, como o *Root Squared Mean Error* (RSME).

Referências

Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, pages 107–144. Springer.

Schwartz, B. (2004). *The Paradox of Choice: Why More Is Less*. Ecco.

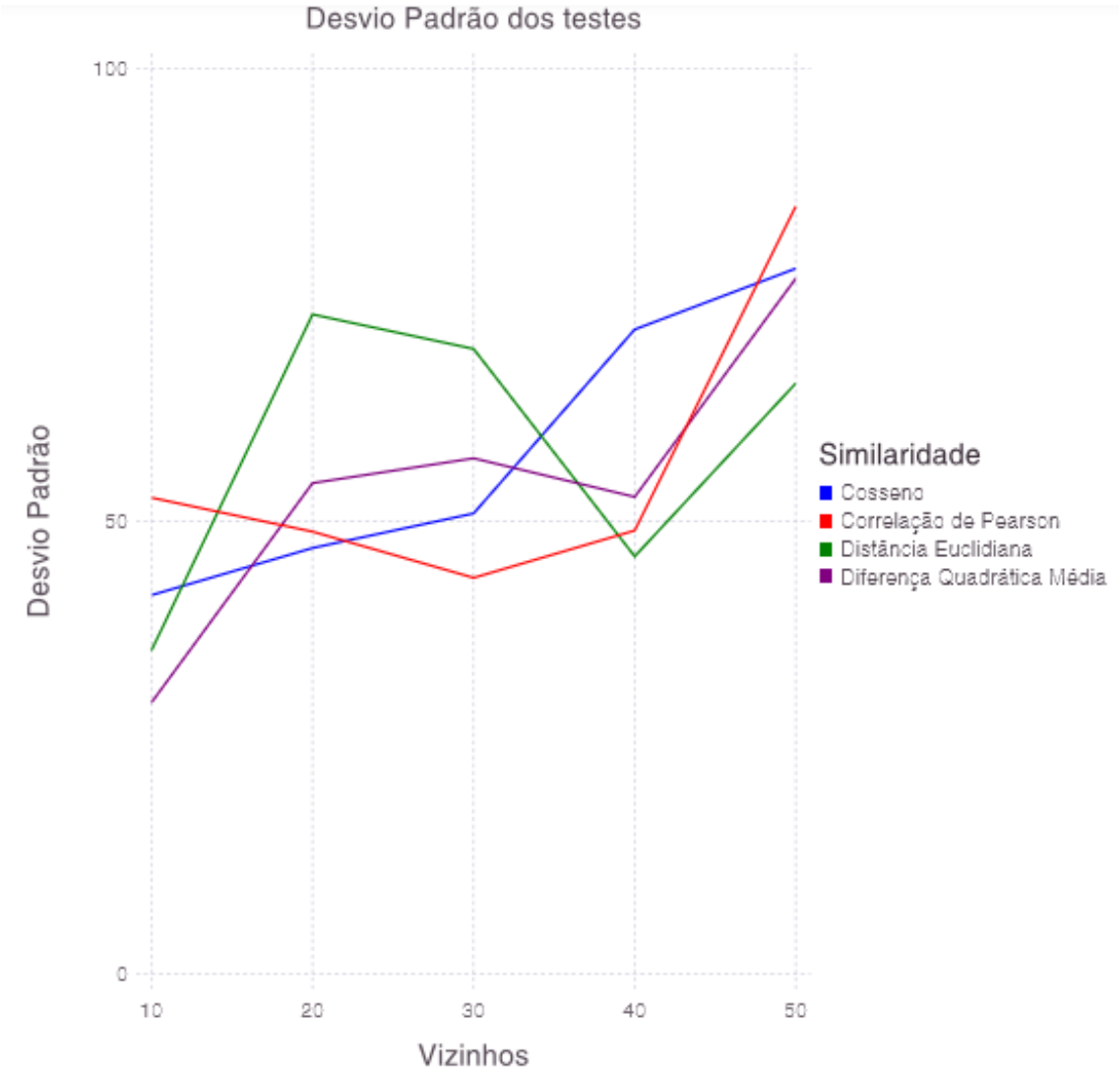

Figura 3. Gráfico das médias dos valores do Desvio Padrão da média da quantidade de elementos que não foram previstos