Leonardo Filgueira

Sistemas de recomendação usando o software R

Niterói - RJ, Brasil

Universidade Federal Fluminense

Leonardo Filgueira

Sistemas de recomendação usando o software R

Trabalho de Conclusão de Curso

Monografia apresentada para obtenção do grau de Bacharel em Estatística pela Universidade Federal Fluminense.

Orientador: Prof. Luciane Ferreira Alcoforado

Niterói - RJ, Brasil

Universidade Federal Fluminense

Leonardo Filgueira

Sistemas de recomendação usando o software R

Monografia de Projeto Final de Graduação sob o título *"Sistemas de recomendação usando o software R"*, defendida por Leonardo Filgueira e aprovada em , na cidade de Niterói, no Estado do Rio de Janeiro, pela banca examinadora constituída pelos professores:

Profa. Dra. Luciane Ferreira Alcoforado
Departamento de Estatística – UFF

Prof. Dr. Steven Dutt Ross
UNIRIO

Prof. Dr. Rodrigo Otávio de Araújo Ribeiro
UERJ

Niterói,

Resumo

No ambiente digital, como lojas, serviços de *streaming*, redes sociais, a sugestão de produtos, conteúdos, pessoas com quem se conectar, é realizada a todo momento, com base em informações dos diversos usuários e itens do site. A recomendação pode ser uma medida que facilita a navegação e experiência do usuário, além de potencialmente aumentar a fidelização dos clientes e o faturamento da empresa.

Os sistemas de recomendação utilizam bases com grande volume de dados, o que pode ser um desafio para o seu processamento, e dividir a base em outras menores pode ser uma maneira de contornar o problema do processamento, além de ser uma possibilidade para atingir uma melhor performance das recomendações. Além de descrever alguns métodos de recomendação, este trabalho aplicará técnicas de clusterização sobre os usuários para comparar a acurácia e o tempo de processamento da recomendação de filmes para usuários.

Palavras-chaves: Sistemas de recomendação, filtragem colaborativa

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

A partir do aumento de informação disponível com a popularização da Internet e com a possibilidade de armazenar essas informações, surge o desafio de lidar com este grande conjunto de dados[1]. Este aumento de informações desafia o site, como lojas on-line, que recebe todas as informações dos usuários que visitam o endereço, mas também pode se tornar um problema para o usuário que, diante da grande quantidade de produtos disponíveis para compra, pode levar muito tempo para achar o produto desejado[2].

Sistemas de recomendação são técnicas de *machine learning* (aprendizado de máquina) que filtram um grande conjunto de dados, tendo como base informações dos usuários e itens[3]. A partir dessas técnicas são previstas as notas que os usuários dariam a determinados itens, que podem ser dos mais variados tipos, e, para um indivíduo, recomenda-se o(s) item(ns) que obtiveram uma nota prevista maior[4]. Os sistemas de recomendação têm como objetivo recomendar itens que interessariam aos usuários[5], beneficiando o usuário e a loja, pois eles aumentam o desempenho da loja, fazendo-a vender uma quantidade maior de produtos, e também facilitam a procura do usuário fazendo-o achar o(s) produto(s) desejado(s) em um menor tempo[1].

O primeiro sistema de recomendação foi criado na década de 90 e tinha como nome "filtragem colaborativa", pois o sistema funcionava com base na colaboração entre os grupos de pessoas interessados. Contudo, o termo "sistemas de recomendação" é mais usado por ser mais geral, não sendo realizada, necessariamente, nenhuma colaboração entre pessoas[6]. Já em 1996 o *Yahoo* utilizou sistemas de recomendação em uma de suas páginas, aplicando em larga escala[6], coisa que hoje é feita comumente por diversos sites e serviços.

É facilmente perceptível no cotidiano o uso de sistemas de recomendação em ambientes on-line. Ao usar a *Netflix*, sugestões para o usuário são oferecidas, baseadas nas atrações já assistidas e/ou avaliadas. Sites de compras como a *Amazon* também oferecem sugestões de produtos ao usuário baseado em visitas à página dos produtos ou no comportamento de

outros usuários que compraram um mesmo produto. Também em redes sociais, como no *YouTube*, são sugeridos vídeos baseados no histórico do internauta e nas suas avaliações, ou então no *Facebook*, que recomenda lista de pessoas que o usuário pode conhecer[7].

Em geral, sistemas de recomendação utilizam como informação a avaliação (*rating*) dada pelos usuários aos itens, podendo a avaliação estar expressa de diferentes maneiras[4]:

- Avaliações numéricas: O usuário avalia um item numa escala numérica, como no site da *Amazon*, onde o usuário dá uma nota de até 5 estrelas.
- Avaliações qualitativas: A avaliação é dada por frases definidas, como: "Concordo totalmente", "Concordo parcialmente", ...
- Avaliações binárias: O usuário seleciona se gostou ou não gostou do item, como a *Netflix*, atualmente, recebe as avaliações.
- Avaliação unária: A indicação se refere a se o usuário visualizou, comprou ou então avaliou o item positivamente.

1.1 Técnicas de recomendação

Existem diferentes categorias de sistemas de recomendação, que podem ser classificados em: Filtragem baseada em conteúdo (*Content-based filtering*), filtragem colaborativa (*Collaborative filtering*) e sistemas de recomendação híbridos (*Hybrid Recommender Systems*)[5].

1.1.1 Filtragem baseada em conteúdo

Os sistemas nesta categoria recomendam itens similares aos que o usuário gostou no passado[7]. Para isto é necessário utilizar informações das características de um produto[4] e comparar com o perfil do usuário, de acordo com itens já conhecidos pelo usuário. Considerando filmes como itens, se um usuário avaliou positivamente filmes do gênero de ação, então o sistema recomendará a este usuário filmes de ação. Por outro lado, a filtragem baseada em conteúdo não leva em conta a similaridade de preferência entre os usuários, mas apenas o histórico do usuário e as características dos itens[7].

Algumas das técnicas utilizadas neste tipo de filtragem são: TF/IDF (*Term Frequency Inverse Document*), *naive Bayes Classifier*, árvores de decisão ou redes neurais[1].

1.1.2 Filtragem colaborativa

Na filtragem colaborativa são recomendados itens de acordo com as avaliações de todos os usuários[5]. Existem duas maneiras principais de realizar essa filtragem: baseado em memória ou em modelo[8]. Nos algoritmos baseados em memória, verifica-se a similaridade entre usuários ou entre itens (vizinhança), de acordo com suas avaliações passadas. Essa técnica é a mais utilizada para realizar recomendações[4]. Um exemplo simples seria: Se o usuário 1 comprou o item A, B e C, e o usuário 2 comprou os itens A e C, então recomenda-se o item B para o usuário 2.

Os algoritmos de filtragem colaborativa utilizam uma matriz, chamada de matriz de avaliações (*ratings matrix*), usualmente representada como na tabela 1.

Tabela 1: Típica matriz **R** de avaliações

	Item 1	Item 2	$\cdots$	Item m
Usuário 1	$r_{(1,1)}$		$\cdots$	
Usuário 2		$r_{(2,2)}$	$\cdots$	$r_{(2,m)}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Usuário n			$\cdots$	$r_{(n,m)}$

Onde $r_{(i,j)}$ é a avaliação (*rating*) do usuário i dado ao item j. Em geral, os usuários não tiveram contato com todos os itens, então os itens não recebem avaliações de todos os usuários, produzindo então uma matriz esparsa (com grande quantidade de valores faltantes). Os algoritmos buscam, então, preencher a matriz de avaliações com previsões para os valores faltantes.

À medida, porém, que os números de usuários e items aumentam, podem surgir problemas ao realizar a filtragem, como o aumento do tempo necessário, além de recursos computacionais, para executar o algoritmo, chamado de problema de escalabilidade[8]. Além disso, existe o problema da esparsidade, pois um usuário, em geral, não avaliou uma grande quantidade de itens, mas apenas uma pequena quantidade, o que pode causar a impossibilidade do cálculo de medidas de similaridade (pois itens precisam ter sido avaliados por dois usuários), ou então pode levar, pela pequena quantidade de informação utilizada no cálculo da medida, a uma medida que não represente bem a real similaridade entre os usuários[8].

Buscando reduzir o tempo de processamento e melhores medidas de acurácia podem ser utilizados métodos de agrupamento (cluster)[9]. Uma possibilidade é agrupar usuários, de acordo com alguma informação disponível em k clusters e, para cada um dos grupos

de usuário, aplicar a técnica de recomendação.

1.1.3 Sistemas de recomendação híbridos

Os sistemas híbridos são uma combinação da filtragem baseada em conteúdo e filtragem colaborativa, buscando aproveitar as vantagens e eliminar as desvantagens das técnicas[4]. Cada uma das técnicas podem ser aplicadas de maneira separada, combinando os resultados, mas também pode ser construído um modelo com as duas abordagens unificadas[3].

2 Objetivos

Este trabalho tem os seguintes objetivos:

2.1 Objetivo geral

Comparar a acurácia das recomendações utilizando filtragem colaborativa para todo o conjunto de dados com as recomendações utilizando filtragem colaborativa para cada cluster de usuários.

2.2 Objetivos específicos

- Descrever informações de filmes e usuários.
- Analisar filmes avaliados e recomendados para determinado usuário.
- Comparar o tempo de execução da recomendação para as configurações escolhidas.

3 Materiais e Métodos

3.1 Conjunto de dados

Será utilizado um *dataset* disponível no site *grouplens*, disponível em `https://grouplens.org/datasets/movielens/1m/`. O conjunto de dados possui 1 000 209 avaliações de 3900 filmes dados por 6040 usuários[10], que se cadastraram no site *MovieLens* no ano de 2000. De acordo com o próprio site, pessoas podem se inscrever para avaliar filmes e receber recomendações de filmes para assistir.

Os usuários são representados pelo seu ID, que varia entre 1 e 6040 e os filmes possuem ID entre 1 e 3952. As avaliações têm formato numérico, de até 5 estrelas, com estrelas completas, tendo cada usuário avaliado ao menos 20 filmes.

A base de dados será dividida em duas, treino e teste, na proporção de 70% para treinar o modelo e 30% que serão usados para que o modelo preveja as notas a fim de comparar com a nota real.

Será utilizada uma segunda base, que apresenta informações sobre os filmes, como o código, nome e gêneros do filme. Um mesmo filme pode ter sido associado a mais de um gênero, mas nenhum filme não foi associado a algum dos 18 gêneros existentes.

Executar a tarefa de recomendação é dificultada a medida em que o tamanho da base de dados aumenta, e fazê-la sem utilizar um servidor, com uma quantidade maior de memória e processamento exige que não se utilize uma base maior. Devido a essa limitação, utilizou-se a base escolhida.

3.2 Metodologia

Haverão um conjunto de usuários $U = \{u_1, u_2, \ldots, u_n\}$ e um conjunto de itens $I = \{i_1, i_2, \ldots, i_m\}$, assim como as notas dos usuários aos itens, que serão armazenadas na matriz $\boldsymbol{R}_{n \times m}$ de avaliações[11]. Logo, cada linha da matriz $\boldsymbol{R}$ representa um usuário e

cada coluna, um item. Os algoritmos buscarão preencher os valores faltantes desta matriz, com valores na mesma escala das avaliações presentes na matriz[3].

3.2.1 Filtragem colaborativa baseada no item (item-based)

Este algoritmo busca recomendar itens similares aos bem avaliados pelo usuário. Desta forma será verificado, para cada par de itens, a sua similaridade, e a partir desta medida é prevista a avaliação do usuário para o item. A similaridade entre dois itens i e j pode ser medida pelo coeficiente de correlação de Pearson, definido da seguinte maneira[5]:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \overline{r}_i)(r_{u,j} - \overline{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \overline{r}_i)^2 \sum_{u \in U} (r_{u,j} - \overline{r}_j)^2}} \tag{3.1}$$

Sendo U o conjunto de usuários que avaliaram os dois itens, i e j,$r_{u,i}$ o rating dado pelo usuário u ao item i e $\overline{r}_i$ o rating médio recebido pelo item i dado por todos os usuários que o avaliaram.

Alternativamente, a similaridade entre os itens i e j pode ser medida considerando os ratings recebidos pelos dois itens como vetores e calcular o cosseno entre estes vetores[12]:

$$w_{i,j} = \cos(\vec{r_i}, \vec{r_j}) = \frac{\vec{r_i} \cdot \vec{r_j}}{\|\vec{r_i}\| \times \|\vec{r_i}\|} = \frac{\sum_{u=1}^{n} r_{u,i} r_{u,j}}{\sqrt{\sum_{u=1}^{n} r_{u,i}^2 \sum_{u=1}^{n} r_{u,j}^2}} \tag{3.2}$$

A seguir, o *rating* do item i pelo usuário a pode ser previsto da seguinte forma[5]:

$$p_{a,i} = \frac{\sum_{j \in k} r_{a,i} - w_{i,j}}{\sum_{j \in k} |w_{i,j}|} \tag{3.3}$$

Sendo k o conjunto de itens avaliados pelo usuário a que são mais similares ao item i.

3.2.2 Filtragem colaborativa baseada no usuário (user-based)

Este algoritmo assume que usuários com preferência similar no passado terão preferências similares no futuro. Então os *ratings* não observados serão previstos a partir das avaliações de uma vizinhança e usuários com gostos similares[11]. São então encontrados os k vizinhos mais próximos de um usuário ou então todos os usuários que tenham pelo menos uma dada similaridade. O coeficiente de correlação de Pearson pode ser utilizado

como medida de similaridade entre dois usuários a e u, definida da seguinte maneira[5]:

$$w_{a,u} = \frac{\sum_{i \in I} (r_{a,i} - \overline{r}_a)(r_{u,i} - \overline{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \overline{r}_a)^2 \sum_{i \in I} (r_{u,i} - \overline{r}_u)^2}} \tag{3.4}$$

Sendo I o conjunto de itens avaliados pelos dois usuários, $r_{u,i}$ é o rating dado pelo usuário u ao item i e $\overline{r}_u$ é o rating médio do usuário u a todos os itens por ele avaliados.

Uma outra maneira de calcular a similaridade entre dois usuários é considerar os ratings de dois usuários como vetores num espaço m-dimensional, para, assim, encontrar o cosseno do ângulo entre estes vetores[5]:

$$w_{a,u} = \cos(\vec{r}_a, \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\| \times \|\vec{r}_u\|} = \frac{\sum_{i=1}^{m} r_{a,i} r_{u,i}}{\sqrt{\sum_{i=1}^{m} r_{a,i}^2 \sum_{i=1}^{m} r_{u,i}^2}} \tag{3.5}$$

Por fim, a predição da nota dada ao item i pelo usuário a é dada por:

$$p_{a,i} = \overline{r}_a + \frac{\sum_{u \in k} (r_{u,i} - \overline{r}_u) w_{a,u}}{\sum_{u \in k} |w_{a,u}|} \tag{3.6}$$

Sendo k a vizinhança do usuário a.

3.2.3 *PAM* (Partitioning Around Medoids)

O algoritmo de agrupamento *PAM* é baseado na definição de *medoide*, que é o ponto com menor distância, em média, de todos os outros elementos do cluster. O algoritmo, para obter k clusters, é executado da seguinte maneira[13]:

1. Definir aleatoriamente k medoides.
2. Associar cada um dos elementos restantes a um cluster, sendo pertencente ao grupo de medoide mais próximo.
3. Calcular a dissimilaridade entre um elemento x_i e todos os outros do cluster, e a dissimilaridade entre o medoide e os outros elementos do cluster.
4. Caso a distância considerando x_i como novo medoide seja menor que a distância do medoide atual, passe a considerar x_i como medoide daquele cluster.
5. Repetir os passos 2 a 4 até não haver troca de medoides.

Uma desvantagem desse método é a ineficiência ao ser aplicado para um grande conjunto de dados[14].

3.2.4 *CLARA* (Clustering Large Applications)

Essa técnica foi proposta, em 1990, de forma a aplicar o PAM, solucionando o problema de escalabilidade, ao utilizar amostragem para a aplicação do PAM[14]. O método, então, seleciona aleatoriamente uma parte da base de dados e aplica o algoritmo PAM nesta amostra. Em seguida é calculada a função de custo, que é uma média da similaridade entre os medoides e os outros elementos da base[15]. A função de custo é definida da seguinte maneira:

$$C(m, D) = \frac{\sum_{i=1}^{n} d(x_i, cl(m, x_i))}{n} \tag{3.7}$$

Onde:

- m são os medoides encontrados.
- $cl(m, x_i)$ é o medoide mais próximo de um ponto x_i.
- $d(x_i, cl(m, x_i))$ é uma medida de similaridade entre x_i e seu medoide mais próximo.
- n é o número de observações na base de dados D.

Todo o processo é repetido um número determinado de vezes e o resultado que obtiver menor função de custo é definido então como o melhor e é retornado[15].

3.2.5 *K-Means*

K-means é uma técnica que particiona elementos em k clusters utilizando-se de centroides, que são os elementos representativos de cada cluster. Este método busca minimizar a soma das distâncias dos elementos de um mesmo cluster. Dados então, uma matriz D, de dimensão $m \times n$, e um número de clusters k, o algoritmo, então, procede da seguinte maneira[16]:

1. São escolhidos, aleatoriamente, k objetos de D como sendo os centroides.
2. Cada elemento $D_{i,}$ é associado ao centroide mais próximo, de acordo com a medida de distância adotada (neste caso, a distância Euclidiana).

3. Os centroides de cada um dos clusters são calculados.

4. Repetir os passos 2 e 3 até que não haja mudanças.

3.2.6 Informações utilizadas para agrupamento

Com o intuito de agrupar os usuários em clusters, foram utilizadas duas bases de dados: a avaliação média dos usuários para cada categoria e a proporção de filmes assistidos pelos usuários para cada categoria. No primeiro caso, utilizando o rating médio, houveram casos em que alguns gêneros não receberam nenhuma nota, gerando um dado faltante. Para preencher os valores faltantes foi usada a média de notas por categoria.

Para que tivessem sido geradas algumas possibilidades de resultados utilizando clusterização, os usuários foram agrupados desde em 2 clusters até em 15 grupos para cada algoritmo e para cada base: Rating médio e proporção de filmes assistidos por categoria.

3.2.7 Medidas de erro

Para verificar a acurácia do sistema de recomendação, dado que a base utilizada foi dividida em base de treino e de teste, serão comparadas a avaliação prevista e a avaliação observada. Considerando $r_{i,j}$ a avaliação observada e $p_{i,j}$ a avaliação prevista pelo modelo do usuário i ao item j, as medidas utilizadas serão[7]:

3.2.7.1 Erro médio absoluto

O erro médio absoluto (EMA) se dá pela soma do módulo das diferenças.

$$EMA = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} |r_{i,j} - p_{i,j}| \tag{3.8}$$

3.2.7.2 Erro quadrático médio

O erro quadrático médio (EQM) é a soma das diferenças ao quadrado. Por este motivo, a unidade de medida muda, e a sua interpretação deve ser cautelosa.

$$EQM = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} (r_{i,j} - p_{i,j})^2 \tag{3.9}$$

3.2.7.3 Raiz do erro quadrático médio

Ao calcular a raiz do EQM obtém-se um número na mesma unidade de medida dos dados.

$$REQM = \sqrt{\frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(r_{i,j} - p_{i,j}\right)^2} \tag{3.10}$$

3.2.8 Software utilizado

A fim de executar todo o processo de leitura e descrição da base, clusterização e recomendação, foi utilizada a linguagem de progrmação R[17], por meio do ambiente de desenvolvimento RStudio[18]. O pacote do R *recommenderlab*[19] executa a recomendação e sua avaliação, produzindo as medidas de erro.

4 Análise dos Resultados

Os 6040 usuários avaliaram pelo menos 20 filmes. Na figura 1, é possível notar que a maior parte dos usuários avaliou até 500 filmes. Além disso, nota-se que essa distribuição apresenta uma assimetria a direita.

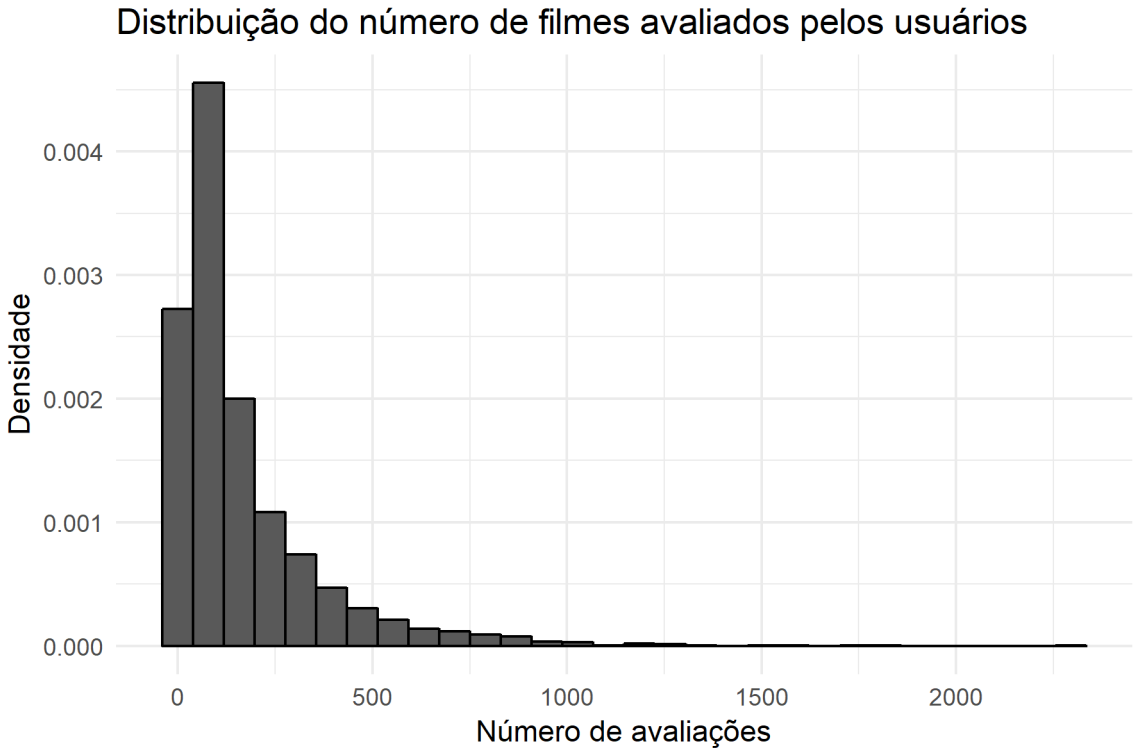

Figura 1: Número de filmes avaliados

A tabela 2 apresenta algumas medidas resumo a respeito da quantidade de filmes avaliados pelos usuários. Nota-se uma grande amplitude, variância, desvio padrão e coeficiente de variação, o que indica uma grande variabilidade na quantidade de filmes avaliados. Metade dos usuários avaliou até 96 filmes. Além disso, 75% dos usuários avaliaram 208 filmes, o que indica, como já foi indicado na figura 1, que um número pequeno de usuários, diferentemente do comportamento da maior parte, avaliou uma grande quantidade de filmes.

Tabela 2: Medidas resumo da quantidade de filmes avaliados

Min.	**Mediana**	**Média**	**Max.**	**Variância**	**Desvio padrão**	**Coef. de variação**
20	96	165.6	2314	37151	193	1.16

A seguir, a tabela 3 apresenta os 20 filmes com maior número de avaliações recebidas. Destaca-se a trilogia original de *Star Wars*, cujos filmes receberam, entre si, uma quantidade muito próxima de avaliações dos usuários.

Tabela 3: Filmes com mais avaliações recebidas

Filme	**Número de avaliações**
American Beauty (1999)	3428
Star Wars: Episode IV - A New Hope (1977)	2991
Star Wars: Episode V - The Empire Strikes Back (1980)	2990
Star Wars: Episode VI - Return of the Jedi (1983)	2883
Jurassic Park (1993)	2672
Saving Private Ryan (1998)	2653
Terminator 2: Judgment Day (1991)	2649
Matrix, The (1999)	2590
Back to the Future (1985)	2583
Silence of the Lambs, The (1991)	2578
Men in Black (1997)	2538
Raiders of the Lost Ark (1981)	2514
Fargo (1996)	2513
Sixth Sense, The (1999)	2459
Braveheart (1995)	2443
Shakespeare in Love (1998)	2369
Princess Bride, The (1987)	2318
Schindler's List (1993)	2304
L.A. Confidential (1997)	2288
Groundhog Day (1993)	2278

A distribuição da quantidade de avaliações recebidas pelos filmes na base de dados é apresentada na figura 2. 114 filmes receberam apenas uma avaliação, 50% dos filmes receberam 124 *ratings* e um filme recebeu 2858 avaliações de usuários. A variabilidade de ratings recebidos pelos filmes é alta, com um coeficiente de variação igual a 1.42.

A tabela 4 apresenta a quantidade de filmes aos quais cada gênero foi atribuído. Como cada filme pode ter sido descrito com mais de um gênero, a soma das frequências é maior que o número de filmes. Nota-se que os gêneros aos quais mais filmes foram associados são drama e comédia. O terceiro gênero com mais filmes associados, ação, apresenta menos de metade do número de filmes, em relação aos dois primeiros.

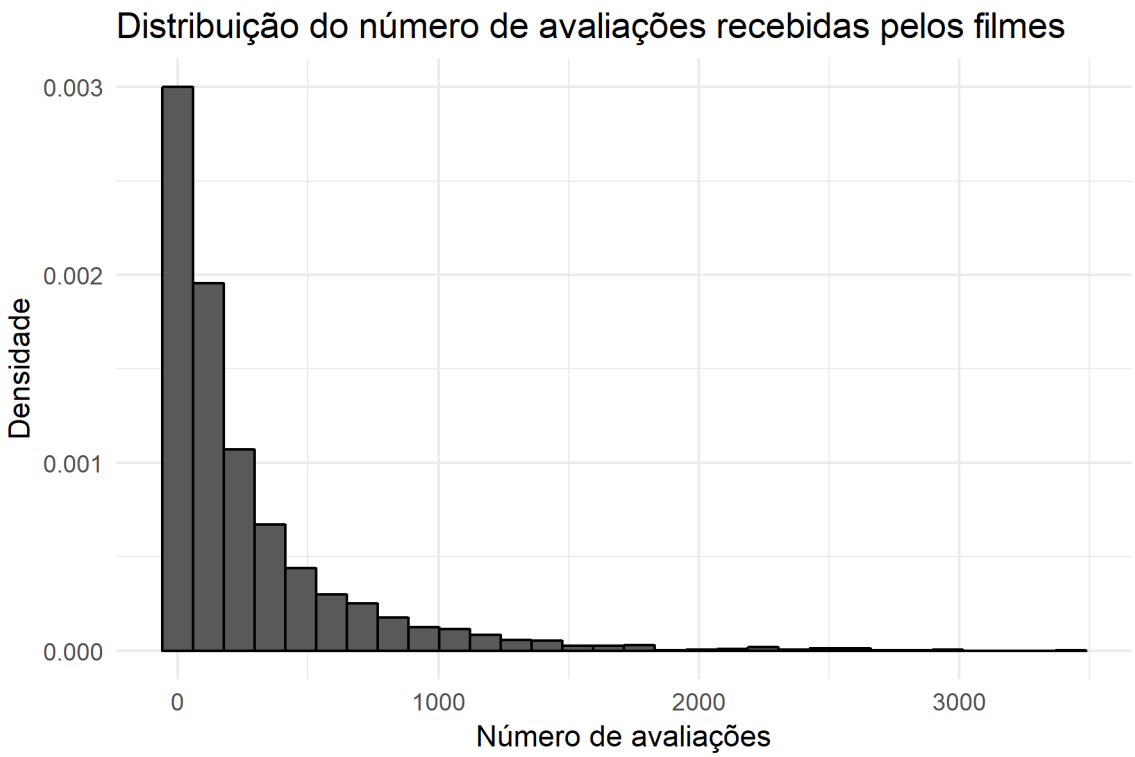

Figura 2: Número de avaliações recebidas pelos filmes

Tabela 4: Gêneros existentes na base e número de filmes associados

Gênero	**Número de filmes**
Action	503
Adventure	283
Animation	105
Children's	251
Comedy	1200
Crime	211
Documentary	127
Drama	1603
Fantasy	68
Film-Noir	44
Horror	343
Musical	114
Mystery	106
Romance	471
Sci-Fi	276
Thriller	492
War	143
Western	68

A figura 3 apresenta as distribuições de nota média recebida para os gêneros presentes na base. A mediana das avaliações médias encontra-se entre 3 e 4 estrelas, com apenas os gêneros *War* (guerra), *Film-Noir* (uma espécie de filme policial) e *Documentary* (documentário) atingindo uma mediana igual a 4. As medianas mais baixas encontram-se em *Horror* e *Sci-Fi* (Ficção científica), com 3.38 e 3.58 como medianas, respectivamente.

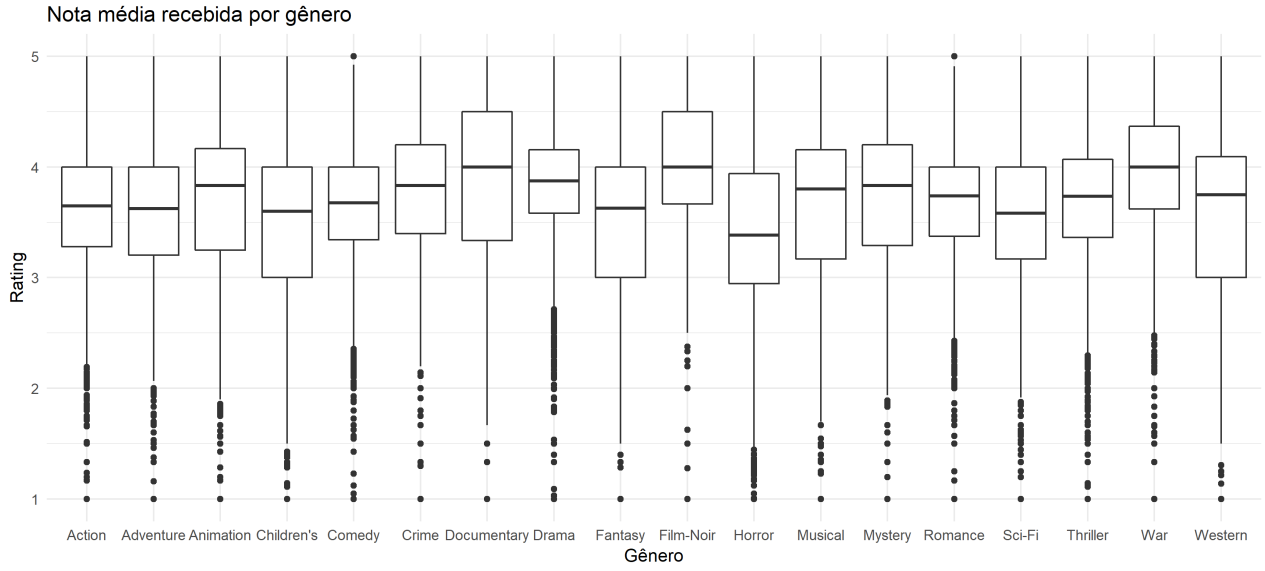

Figura 3: Distribiuição da avaliação médio por gênero

A seguir será verificada a recomendação para um usuário em específico, que avaliou 20 filmes. A seleção da pessoa foi feita aleatoriamente. Primeiramente devem ser apresentadas as avaliações do usuário aos filmes, para comparar com os filmes que seriam recomendados para ele. A tabela 5 apresenta essa informação.

Tabela 5: Avaliações do usuário 341

Filme	**Gênero**	**Avaliação**
Nikita (La Femme Nikita) (1990)	Thriller	5
Mission: Impossible (1996)	Action, Adventure, Mystery	5
Somewhere in Time (1980)	Drama, Romance	5
East of Eden (1955)	Drama	5
Braveheart (1995)	Action, Drama, War	5
Hard-Boiled (Lashou shentan) (1992)	Action, Crime	5
Out of Sight (1998)	Action, Crime, Romance	5
American Beauty (1999)	Comedy, Drama	5
Airplane! (1980)	Comedy	5

Tabela 5: Avaliações do usuário 341

Filme	**Gênero**	**Avaliação**
Boat, The (Das Boot) (1981)	Action, Drama, War	5
Contact (1997)	Drama, Sci-Fi	4
Frequency (2000)	Drama, Thriller	4
Superman (1978)	Action, Adventure, Sci-Fi	4
Tank Girl (1995)	Action, Comedy, Musical, Sci-Fi	4
Alien (1979)	Action, Horror, Sci-Fi, Thriller	4
Pitch Black (2000)	Action, Sci-Fi	3
Shanghai Noon (2000)	Action	3
Run Lola Run (Lola rennt) (1998)	Action, Crime, Romance	3
Jurassic Park (1993)	Action, Adventure, Sci-Fi	3
Perfect Storm, The (2000)	Action, Adventure, Thriller	2

De acordo com a tabela 5, nota-se que a maior parte dos filmes avaliados é do gênero de ação, porém alguns receberam avaliações muito boas, de 5 estrelas, enquanto alguns receberam apenas 3 ou até mesmo 2 estrelas. Os filmes que contém comédia, drama ou guerra em geral receberam boas notas, com, pelo menos 4 estrelas.

Ao executar o sistema de recomendação, selecionando as 10 maiores avaliações previstas, tem-se uma recomendação de 10 filmes para esse usuário. A tabela 6 apresenta o que seria a recomendação dos filmes.

Tabela 6: 10 maiores notas previstas para o usuário 341

Filme	**Gênero**	**Avaliação**
Pulp Fiction (1994)	Crime, Drama	4.55
Schindler's List (1993)	Drama, War	4.55
Casablanca (1942)	Drama, Romance, War	4.52
Sixth Sense, The (1999)	Thriller	4.51
L.A. Confidential (1997)	Crime, Film-Noir, Mystery, Thriller	4.51
Gladiator (2000)	Action, Drama	4.49
Being John Malkovich (1999)	Comedy	4.48
Saving Private Ryan (1998)	Action, Drama, War	4.48
Godfather, The (1972)	Action, Crime, Drama	4.47

Tabela 6: 10 maiores notas previstas para o usuário 341

Filme	**Gênero**	**Avaliação**
Shakespeare in Love (1998)	Comedy, Romance	4.47

A recomendação, levando em conta os gêneros, é aceitável, pois verifica-se notas altas previstas a filmes de drama, guerra, comédia, e até ação. Os três primeiros gêneros receberam apenas avaliações boas pelo usuário, mas o último recebeu avaliações positivas e também negativas, mas muitos dos filmes avaliados eram desse gênero, o que pode indicar algum interesse do usuário por este tipo de filme. Além disso, destaca-se o filme *L.A. Confidential*, que é associado ao gênero *Film-Noir*, sendo próximo de filmes de ação ou crime.

A seguir serão apresentados a raiz do erro quadrático médio, o erro quadrático médio e erro médio absoluto entre avaliação prevista e observada. O número de clusters foi variado entre 2 e 15, utilizando as técnicas CLARA e k-means. Como a tabela 7 apresenta, a instância "Sem clusterização"é a recomendação executada sem que os usuários fossem agrupados. A tabela está ordenada de acordo com a raiz do EQM, em ordem crescente.

Tabela 7: Medidas de erro considerando as várias configurações

Método	**Informação utilizada**	**Clusters**	**Raiz do EQM**	**EQM**	**EMA**
CLARA	Rating	3	1.0055	1.0199	0.7983
k-means	Proporção	12	1.0213	1.0444	0.8093
k-means	Proporção	8	1.024	1.0495	0.8162
k-means	Rating	12	1.0251	1.052	0.817
CLARA	Rating	4	1.0254	1.0634	0.8188
k-means	Proporção	15	1.0266	1.055	0.8182
k-means	Rating	11	1.0278	1.0568	0.8156
CLARA	Proporção	5	1.0288	1.0599	0.816
k-means	Rating	15	1.03	1.0624	0.8219
CLARA	Proporção	7	1.0315	1.0668	0.825
k-means	Proporção	14	1.0325	1.0667	0.8191
k-means	Rating	13	1.0328	1.0678	0.8193
k-means	Proporção	6	1.0332	1.0682	0.8225

Tabela 7: Medidas de erro considerando as várias configurações

Método	**Informação utilizada**	**Clusters**	**Raiz do EQM**	**EQM**	**EMA**
k-means	Proporção	13	1.0333	1.0685	0.823
k-means	Rating	6	1.0338	1.0688	0.8185
	Sem clusterização		**1.0341**	**1.0693**	**0.8221**
CLARA	Proporção	12	1.0348	1.0742	0.8215
k-means	Rating	14	1.0351	1.0723	0.8236
k-means	Proporção	10	1.0358	1.074	0.8237
CLARA	Proporção	6	1.0367	1.0764	0.8231
k-means	Rating	9	1.0379	1.0775	0.8285
k-means	Proporção	9	1.0387	1.0795	0.8273
k-means	Proporção	4	1.0397	1.0814	0.8286
k-means	Rating	3	1.0399	1.0815	0.8278
k-means	Rating	8	1.041	1.084	0.83
k-means	Rating	7	1.0413	1.0846	0.8287
k-means	Proporção	2	1.042	1.0858	0.8295
CLARA	Proporção	4	1.0422	1.0871	0.8306
k-means	Proporção	11	1.0425	1.0884	0.8302
k-means	Rating	10	1.0432	1.0888	0.8322
k-means	Proporção	3	1.0443	1.0907	0.8345
k-means	Rating	5	1.0446	1.0914	0.8351
CLARA	Proporção	3	1.0447	1.0921	0.8301
CLARA	Proporção	8	1.0452	1.1013	0.8348
k-means	Rating	4	1.0462	1.0946	0.8351
CLARA	Proporção	15	1.0468	1.1028	0.8378
k-means	Rating	2	1.0476	1.0975	0.8378
CLARA	Proporção	2	1.0476	1.0976	0.835
k-means	Proporção	7	1.0518	1.1075	0.8368
CLARA	Proporção	9	1.0585	1.1221	0.8452
k-means	Proporção	5	1.0602	1.1242	0.8407
CLARA	Proporção	11	1.0639	1.1374	0.8448
CLARA	Proporção	10	1.0734	1.1555	0.8576
CLARA	Proporção	14	1.0761	1.1611	0.8614

Tabela 7: Medidas de erro considerando as várias configurações

Método	**Informação utilizada**	**Clusters**	**Raiz do EQM**	**EQM**	**EMA**
CLARA	Proporção	13	1.0762	1.1642	0.8573
CLARA	Rating	2	1.0827	1.1792	0.8711
CLARA	Rating	6	1.0894	1.2354	0.8881
CLARA	Rating	13	1.0948	1.2513	0.8915
CLARA	Rating	10	1.1087	1.2502	0.9061
CLARA	Rating	9	1.1091	1.2444	0.8942
CLARA	Rating	14	1.1165	1.3091	0.9202
CLARA	Rating	11	1.1167	1.2909	0.9197
CLARA	Rating	5	1.1194	1.2743	0.9096
CLARA	Rating	7	1.1282	1.299	0.928
CLARA	Rating	8	1.1377	1.3082	0.9302
CLARA	Rating	12	1.1867	1.5236	0.9671
CLARA	Rating	15	1.1926	1.4865	0.9851

Um fato muito notório ao observar as primeiras linhas é que o método CLARA apresentou melhores resultados com um número menor de clusters, de até 5 grupos, enquanto que o K-means apresentou bons resultados com um número maior de grupos, de 8 até 15, número máximo de clusters.

O valor de referência é o erro obtido ao executar a recomendação sem o particionamento da base. O valor mais baixo da raiz do EQM, utilizando a técnica CLARA a partir do rating médio dos usuários aos gêneros, com 3 clusters, de 1.0055 é aproximadamente 3% menor que a mesma medida sem o particionamento dos usuários. Por outro lado, o maior erro também foi atingido através do rating médio, com o algoritmo que executa a técnica CLARA, mas desta vez com 15 clusters.

Apesar de terem sido obtidos 15 resultados melhores, em relação ao atingido com toda a base, mais de 40 resultados, ao clusterizar os usuários, foram ainda piores que o valor de referência, o que indica que o uso de um método de particionamento não garante um melhor resultado.

Por outro lado, ao agrupar os usuários o tempo de processamento diminuiu, como indica a tabela 8, percebe-se uma tendência ao decréscimo do tempo necessário, com uma

diferença de aproximadamente 100 segundos entre a recomendação com a base completa e com 15 clusters, com a técnica CLARA, tendo sido utilizado o rating médio.

Tabela 8: Tempos de execução da recomendação (em segundos)

Método	**Informação**	**Clusters**	**Tempo (s)**
Sem clusterização			**130**
CLARA	Rating	2	84
CLARA	Rating	3	62
CLARA	Rating	4	46
CLARA	Rating	5	46
CLARA	Rating	6	40
CLARA	Rating	7	37
CLARA	Rating	8	42
CLARA	Rating	9	32
CLARA	Rating	10	32
CLARA	Rating	11	32
CLARA	Rating	12	34
CLARA	Rating	13	31
CLARA	Rating	14	33
CLARA	Rating	15	30

5 Conclusão

O trabalho buscou verificar se existia alguma diferença entre a acurácia da filtragem colaborativa, considerando a base completa de avaliações e a divisão dos usuários em clusters, para executar a filtragem dentro de cada grupo. A diferença entre as medidas de erro do valor de referência e da configuração que obteve o menor erro não é tão significante, e, desse pequeno ganho de acurácia, para alguns números de clusters, a maior parte das tentativas de clusterização obtiveram resultado pior, de acordo com as medidas de erro.

A presença de alguns usuários que avaliaram muito filmes, como o caso de um que avaliou 2314 filmes pode ser explicada pelo fato de que o site oferece serviço de recomendação de filmes, e isso pode atrair pessoas que têm hábito de assistir mais filmes do que a maior parte das pessoas. Numa outra situação, o número de itens avaliados pode ser bem menor.

Ao verificar a recomendação de um usuário específico pôde ser constatado que os filmes recomendados não parecem ser completamente ao acaso, aleatórios, mas sim, são de alguma forma similares aos avaliados pelo usuário. Além disso os filmes recomendados foram classificados por gêneros em geral bem avaliados pela pessoa.

Com relação ao tempo de execução, o agrupamento dos usuários foi uma tarefa fácil, não havendo nem um momento de espera pela execução da função pelo software R. Já no momento de executar a recomendação e calcular o erro gerado, um maior tempo foi necessário, além de utilizar uma quantidade razoavelmente grande de memória do computador, considerando um dispositivo de 8Gb de memória.

O tempo gasto no processamento das recomendações, por outro lado, foi consideravelmente menor quando utilizou-se o agrupamento de usuários. Com isso, uma boa escolha de clusters pode ser muito vantajosa, por economizar tempo de processamento e ter maior acertividade. Trabalhos futuros podem buscar associar técnicas para escolha do número de clusters com a acurácia das recomendações.

Referências

[1] ISINKAYE, F.; FOLAJIMI, Y.; OJOKOH, B. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, Elsevier, v. 16, n. 3, p. 261–273, 2015.

[2] MILD, A.; NATTER, M. Collaborative filtering or regression models for internet recommendation systems? *Journal of Targeting, Measurement and Analysis for marketing*, Springer, v. 10, n. 4, p. 304–313, 2002.

[3] TAKAHASHI, M. M.; JR, R. H. Estudo comparativo de algoritmos de recomendação. *USP. São Paulo*, 2015.

[4] SHAPIRA, B. et al. *Recommender systems handbook*. [S.l.]: Springer New York, 2011.

[5] MELVILLE, P.; SINDHWANI, V. Recommender systems. In: *Encyclopedia of machine learning*. [S.l.]: Springer, 2011. p. 829–838.

[6] REATEGUI, E. B.; CAZELLA, S. C. Sistemas de recomendaçao. In: *XXV Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2005. p. 306–348.

[7] GORAKALA, S. K.; USUELLI, M. *Building a recommendation system with R*. [S.l.]: Packt Publishing Ltd, 2015.

[8] DAKHEL, G. M.; MAHDAVI, M. A new collaborative filtering algorithm using k-means clustering and neighbors' voting. In: IEEE. *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on*. [S.l.], 2011. p. 179–184.

[9] O'CONNOR, M.; HERLOCKER, J. Clustering items for collaborative filtering. In: UC BERKELEY. *Proceedings of the ACM SIGIR workshop on recommender systems*. [S.l.], 1999. v. 128.

[10] HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, ACM, v. 5, n. 4, p. 19, 2016.

[11] HAHSLER, M. *recommenderlab: A framework for developing and testing recommendation algorithms*. [S.l.], 2015.

[12] SARWAR, B. et al. Item-based collaborative filtering recommendation algorithms. In: ACM. *Proceedings of the 10th international conference on World Wide Web*. [S.l.], 2001. p. 285–295.

[13] VALE, M. N. do. *Agrupamentos de dados: Avaliação de Métodos e Desenvolvimento de Aplicativo para Análise de Grupos*. Tese (Doutorado) — PUC-Rio, 2005.

[14] PARK, H.-S.; JUN, C.-H. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, Elsevier, v. 36, n. 2, p. 3336–3341, 2009.

[15] BHAT, A. K-medoids clustering using partitioning around medoids for performing face recognition. *International Journal of Soft Computing, Mathematics and Control*, Citeseer, v. 3, n. 3, p. 1–12, 2014.

[16] MINING, W. I. D. Data mining: Concepts and techniques. *Morgan Kaufinann*, 2006.

[17] R Core Team. *R: A Language and Environment for Statistical Computing*. Vienna, Austria, 2018. Disponível em: <https://www.R-project.org/>.

[18] RStudio Team. *RStudio: Integrated Development Environment for R*. Boston, MA, 2016. Disponível em: <http://www.rstudio.com/>.

[19] HAHSLER, M. *recommenderlab: Lab for Developing and Testing Recommender Algorithms*. [S.l.], 2017. R package version 0.2-2. Disponível em: <https://CRAN.R-project.org/package=recommenderlab>.