PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

APRENDIZADO NEURAL DE REPRESENTAÇÃO DE CONTEÚDO PARA SISTEMA DE RECOMENDAÇÃO DE FILMES

RALPH JOSÉ RASSWEILER FILHO

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Rodrigo Coelho Barros

Porto Alegre
2017

Ficha Catalográfica

R228 Rassweiler Filho, Ralph José

Aprendizado Neural de Representação de Conteúdo para Sistemas de Recomendação de Filmes / Ralph José Rassweiler Filho . – 2017.
65 f.
Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Rodrigo Coelho Barros.

1. Sistemas de Recomendação. 2. Redes Neurais Convolucionais. 3. Filtragem Baseada em Conteúdo. 4. Reconhecimento de Padrões. I. Barros, Rodrigo Coelho. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).

Ralph José Rassweiler Filho

Aprendizado Neural de Representação de Conteúdo para Sistemas de Recomendação de Filmes

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre em Ciência da Computação do Programa de Pós-Graduação em Ciencia da Computação, Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul.

Aprovado em 22 de Agosto de 2017.

BANCA EXAMINADORA:

Prof. Dr. Duncan Dubugras Alcoba Ruiz (PUCRS)

Prof. Dr. Leandro Krug Wives (UFRGS)

Prof. Dr. Rodrigo Coelho Barros (PPGCC/PUCRS - Orientador)

"After one has played a vast quantity of notes and more notes, it is simplicity that emerges as the crowning reward of art. Simplicity is the final achievement."
(Frédéric Chopin)

APRENDIZADO NEURAL DE REPRESENTAÇÃO DE CONTEÚDO PARA SISTEMA DE RECOMENDAÇÃO DE FILMES

RESUMO

Sistemas de recomendação são softwares cujo propósito é gerar listas personalizadas, de acordo com as preferências de usuários. A área é bastante recente e está em expansão desde a popularização da internet tendo suas raízes em recuperação de informação. Dos dois tipos tradicionais de sistemas de recomendação, a filtragem colaborativa é a mais utilizada na academia e na indústria por trazer melhores resultados que o segundo tipo, a filtragem baseada em conteúdo. Este último sofre de problemas tais como a falta de informação semântica e a dificuldade em extrair conteúdo dos itens. Atualmente há uma maior disponibilidade de conteúdo de itens na forma de recursos multimídia tais como vídeos, imagens e texto. Também houve avanços no reconhecimento de padrões em imagens através de técnicas como as redes neurais convolucionais. Neste trabalho, propõe-se utilizar uma rede neural convolucional como extratora de atributos dos quadros que compõe trailers de filmes que servem como base para um sistema de recomendação baseado em conteúdo com o objetivo de avaliar se o sucesso destas redes em tarefas como classificação de imagens e detecção de objetos também ocorre no contexto de recomendações. Para esta avaliação, comparou-se o método proposto com um método de detecção de estética de mídia, dois métodos de extração de conteúdo de texto usando TF-IDF e os tradicionais métodos colaborativos entre usuários e itens. Os resultados obtidos mostram que o método proposto neste trabalho é superior aos demais métodos baseados em conteúdo e é competitivo com os métodos colaborativos, superando o método colaborativo entre itens na métrica que representa acurácia de classificação e também, superando todos os outros métodos com relação ao tempo de execução. Concluiu-se que o método que utiliza redes neurais convolucionais para representar itens é promissor para o contexto de sistemas de recomendação.

Palavras Chave: sistemas de recomendação, redes neurais convolucionais, reconhecimento de padrões, filtragem baseada em conteúdo.

NEURAL CONTENT REPRESENTATION LEARNING FOR MOVIE RECOMMENDER SYSTEMS

ABSTRACT

Recommender systems are software used to generate personalized lists according to users profiles. The area is new and is growing since the internet popularization having its roots in information retrieval. Collaborative filtering is the most common approach of recommender systems used in both academy and industry because content-based filtering has problems such as lack of semantic information and poor content extraction techniques from items. Nowadays there are more content available in the form of multimedia such as video, images and text. Also, there are advances in pattern recognition though techniques like convolutional neural networks. In this work a convolutional neural network is used to extract features from movie trailers frames to further use these features to create a content-based recommender system with the goal of assessing whether the success of such networks on tasks like image classification and object detection also occur in the recommendation context. To evaluate that, the proposed method was compared with a media aesthetic detection method, two methods of feature extraction from text using TF-IDF and the traditional user and item collaborative filtering methods. Our results indicate that the proposed method is superior to the other content-based methods and is competitive to the collaborative filtering methods, being superior to the item-collaborative method regarding classification accuracy, and being superior to all other methods regarding execution time. In conclusion, we can state that the method using convolutional neural networks to represent items is promising for the recommender systems context.

Keywords: recommender systems, convolutional neural networks, pattern recognition, content-based filtering, deep learning.

SUMÁRIO

1. INTRODUÇÃO

Sistemas de recomendação (SR) são softwares construídos para montar listas de itens de interesse dos usuários de acordo com seus perfis. Com a popularização da internet a partir da década de 1990, vários *websites* passaram a tirar vantagem deste tipo de software e, consequentemente, a pesquisa científica aumentou ao mesmo tempo que a área de SR foi formalizada, derivada da área de recuperação de informações (*information retrieval*) e diretamente associada às áreas de aprendizado de máquina e mineração de dados.

Um dos domínios mais popularmente aplicados à SR é o de filmes, que recebeu uma atenção especial tanto da academia quanto da indústria em função da disponibilização de *datasets* públicos como o MovieLens (desde 1997) [HK16] e o prêmio Netflix (anunciado em 2006) [BL+07]. No *dataset* MovieLens, mantido pela Universidade de Minnesota, os itens são avaliados pelos usuários em forma de uma nota que variam em uma escala de **0,5** a **5,0**. A versão estável mais recente do MovieLens é um *dataset* com cerca de 250 mil usuários, 30 mil filmes, mais de 20 milhões de avaliações e mais de 500 mil aplicações de *tags*. O prêmio Netflix, publicado por empresa homônima em 2006, prometeu 1 milhão de dólares ao algoritmo que melhorasse em 10% a acurácia de predição de notas medida pela empresa. A empresa disponibilizou um *dataset* contendo 100 milhões de avaliações anônimas. Devido a posteriores tentativas de identificar perfis de usuários com base em avaliações, a Netflix decidiu não disponibilizar mais seu *dataset*. O prêmio foi vencido três anos após o início da competição, por uma equipe que apresentou uma solução resultante de uma combinação linear de cerca de 100 resultados [BKV08].

Os SR podem ser classificados de várias formas, sendo que as duas classificações mais comuns são: filtragem baseada em conteúdo (FBC) e filtragem colaborativa (FC) [LMY+12]. De acordo com Burke [Bur02], o objetivo da FC é identificar usuários ou itens similares a fim de, através apenas das suas avaliações, montar recomendações. Distintamente, o objetivo da FBC é recomendar itens parecidos com os itens que o usuário demonstrou preferência. Para avaliar a similaridade entre os itens, diferentes métricas podem ser utilizadas como por exemplo, o cosseno ou a correlação de Pearson. A FC tem as vantagens de: (a) identificar nichos de gêneros diferentes, ou seja, por mais eclético que seja o usuário, é possível gerar boas recomendações; (b) o domínio do negócio não é necessário; (c) a qualidade da solução melhora com o passar do tempo; (d) o *feedback* implícito do usuário é suficiente. A FBC possui os benefícios (b), (c) e (d). As desvantagens da FC são: (e) falha ao recomendar para novos usuários; (f) falha ao recomendar novos itens; (g) problema da "ovelha negra", ou seja, um usuário com preferências muito particulares; (h) a qualidade depende de um grande *dataset*; (i) problema de estabilidade versus plasticidade, ou seja, se um usuário mudar suas preferências, demorará para o sistema mudar as recomendações. A FBC possui as desvantagens (e), (h) e (i) e além disso possui como desvantagem a falta de serendipidade, o que significa que poucas recomendações surpreendem o usuário devido ao fato de que a FBC recomenda itens similares aos itens que o usuário conhece, um problema denominado superespecialização. Por

fim, a FBC é prejudicada por outro problema, conhecido como análise de conteúdo limitada, pois há certa dificuldade em extrair informações que representem de forma adequada os itens [AT05].

Outra abordagem comum a SR são os modelos híbridos, que combinam características do dois tipos, FC e FBC, e podem ser implementados de várias formas. Muitas aplicações utilizadas hoje em dia, como turismo e serviços financeiros, são complexas e requerem mais esforços para que um sistema de recomendação consiga cumprir seu papel, não bastando executar abordagens tradicionais destes sistemas [AT05]. Conforme descrito em Lü *et al.* [LMY+12], são muitos os desafios para um sistema de recomendação, dentre eles: dispersão dos dados, escalabilidade, escolha das métricas de avaliação e o equilíbrio entre diversidade e acurácia. Para os autores, a ciência da recomendação está apenas no início e, apesar de ter havido grande progresso, há muito espaço para inovação.

Apesar de ser um tipo de sistema de recomendação comum, a FBC é muito pouco utilizado na indústria e pouco explorado na academia. Isso ocorre porque, apesar da FBC utilizar os atributos dos itens, muitas vezes é difícil extrair informações semânticas significativas dos mesmos. O funcionamento básico de um sistema de recomendação FBC está demonstrado na Figura 1.1 [RRS11].

Redes neurais artificiais (RNA), de acordo com Haykin [Hay08], são modelos computacionais que imitam o funcionamento do cérebro humano, tendo obtido grandes avanços nas tarefas de reconhecimento de padrões. Redes neurais convolucionais (RNC) introduzidas por LeCun *et al.* [LBD+90], são um tipo de RNA que tem demonstrado bom desempenho em tarefas que envolvem reconhecimento de padrões em imagens [JLK+15]. Recentemente houve avanços nas tarefas de classificação de imagens e detecção de objetos proporcionados por tais redes [KSH12, HZRS16]. Devido a esse fato este trabalho propõe-se a responder duas questões: (i) é possível melhorar o desempenho de sistemas de recomendação medidos por diferentes métricas de qualidade tais como MAE, F1 e diversidade em comparação a outros métodos baseados em conteúdo e (ii) um método de representação de conteúdo construído a partir de redes neurais convolucionais é competitivo em comparação com os tradicionais métodos colaborativos de sistemas de recomendação.

Este trabalho tem como objetivo principal comparar diferentes abordagens de representação de conteúdo e colaborativas no contexto de sistemas de recomendação de filmes. Em específico, pretende-se avaliar o desempenho de uma representação de conteúdo baseada em redes neurais convolucionais em comparação com uma abordagem baseada em atributos visuais de baixo nível que captura a estética de mídia (*low-level features*) e outras duas abordagens baseadas em texto (com pesos calculados por $TF\text{-}IDF$), bem como duas abordagens colaborativas tradicionais. Para validar cada uma destas abordagens, utilizou-se um *dataset* de filmes, a saber, o LMTD [SWBR16], que contém cerca de 3.500 trailers de filmes em conjunto com um *dataset* de avaliações destes mesmos filmes, o Movielens [HK16], que contém milhões de avaliações de usuários bem como aplicações de *tags* aos filmes e informações como a sinopse de cada filme. Também é escopo deste trabalho avaliar os resultados sobre diferentes aspectos: qualidade de predição (MAE), qualidade das listas de recomendação através das medidas Precisão (*Precision*) e Revocação (*Recall*) sumarizadas

Figura 1.1 – Funcionamento de um sistema de recomendação baseado em FBC: filmes similares aos que o usuário demonstrou preferência possivelmente serão recomendados. O inverso ocorre com filmes que o usuário demonstrou não ter gostado.

por $F1$ e diversidade, esta última para analisar a característica de que um SR baseado em conteúdo possui itens muito similares entre si.

Foram utilizados quatro técnicas de representação de conteúdo bem como duas de representação colaborativa: representação visual de baixo nível (*low-level features*), tendo esta o objetivo de capturar a estética de mídia; representação visual de alto nível (*high-level features*) onde foi utilizado uma arquitetura de RNC (ResNet) com 152 camadas para representar o conteúdo de trailers de filmes; representação textual de alto nível utilizando as sinopses dos filmes; e outra representação textual de alto nível utilizando *tags* aplicadas pelos usuários aos filmes. As representações colaborativas utilizadas foram a colaborativa entre usuários (FC-U) e colaborativa entre itens (FC-I).

Será demonstrado que, apesar de os SRs FBC serem inferiores aos modelos tradicionais que utilizam FC, é possível melhorar os resultados de todas as medidas utilizando hibridização. Foi utilizado o método de hibridização de média aritmética entre as notas preditas.

A contribuição deste trabalho é a criação de um modelo de representação de conteúdo baseado em rede neural convolucional.

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 descreve o que são SR, quais são os tipos, como podem ser implementados e como seu desempenho pode ser avaliado. O Capítulo 3 detalha o método proposto neste trabalho, uma abordagem baseada em redes neurais. O Capítulo 4 descreve a metodologia e os resultados dos experimentos realizados. O Capítulo 5 faz as considerações finais.

2. SISTEMAS DE RECOMENDAÇÃO

Neste capítulo será realizada uma revisão da literatura em sistemas de recomendação. Inicialmente, será detalhado o que são sistemas de recomendação e para que finalidade são utilizados. Serão descritos dois tipos de sistemas de recomendação: filtragem colaborativa, uma técnica que utiliza apenas as avaliações dos usuários de um sistema (Seção 2.2) e a filtragem baseada em conteúdo (Seção 2.3), que é uma técnica que foi inspirada em métodos da área de recuperação de informações, especialmente em motores de buscas. Em seguida, serão detalhados os sistemas de recomendação híbridos (Seção 2.4), que se refere à técnicas de combinação entre os dois filtros citados. Por fim, serão abordadas as formas de avaliação utilizadas para medir o desempenho de sistemas de recomendação (Seção 2.5).

2.1 Conceitos Básicos

Um sistema de recomendação é um software utilizado em contextos onde existem usuários e itens, sendo que o objetivo deste software é estimar quais são os itens que cada usuário tem a maior chance de vir a consumir. Estes itens podem ser filmes, músicas, artigos científicos, notícias, bens e serviços e até mesmo pessoas. Portanto, domínios de *streaming* de vídeos como Netflix, Youtube e Vimeo, *streaming* de músicas como Spotify, Deezer e Pandora, portais de notícias como Globo, Terra e Reuters, portais de *e-commerce* como Wallmart, Americanas e Amazon, redes sociais como Twitter, Facebook e Instagram e portais de serviços e avaliações como Booking, Trip Advisor e Yelp são exemplos de aplicações onde sistemas de recomendação podem desempenhar um papel importante na atração e retenção de clientes.

Citado em Resnick *et al.* [RV97] como sendo o primeiro artigo publicado sobre um sistema de recomendação, Goldberg *et al.* [GNOT92] descreve a solução *Tapestry*, um sistema experimental de envio de e-mails que permitia a personalização dos e-mails recebidos pelos usuários através da aplicação de filtros baseados não apenas no conteúdo dos documentos mas também envolvendo uma avaliação dos leitores a respeito destes documentos. Como exemplo, um usuário poderia filtrar documentos com uma *query* de busca indicando nomes de outros usuários que teriam avaliado de forma positiva notícias de uma *newsletter*. Em seu trabalho, Resnick *et al.* [RV97] descrevem cinco exemplos de aplicações práticas em sistemas de recomendação, contrastando a forma de avaliação, a origem dos dados de *feedback* dos usuários, a possibilidade de anonimato dos usuários, como os itens são avaliados para serem recomendados e, por fim, como as recomendações são enviadas aos usuários. Esses cinco aspectos abordados pelos autores serviram como base para a diversificação de estudos em sistemas de recomendação. Como exemplos, pode-se citar o trabalho relatado em Balabanovic [Bal98] que investiga o *trade-off* entre recomendar itens que o sistema tem segurança para indicar, com base no histórico de avaliações dos usuários em documentos relacionados, e a indicação de novos documentos. Em Lawrence *et al.* [LAK+01] é feito a descrição de um sistema de

recomendação para sugerir produtos para clientes de um supermercado em PDA's (*Personal Digital Assistant*). Em Cohen *et al.* [CSS99] foram abordados aspectos que envolvem a ordenação de itens.

O trabalho de Adomavicius *et al.* [AT05] indica que os sistemas de recomendação podem ser definidos em três categorias: filtragem colaborativa, filtragem baseada em conteúdo e híbridos. Cada um desses tipos será detalhado nesta seção. Os autores definem também quais são as técnicas comumente aplicadas para implementar cada um dos tipos de sistemas de recomendação citados bem como quais são as vantagens e desvantagens de cada um deles. Além disso, foram definidas extensões para os tipos existentes de sistemas de recomendação, como por exemplo, técnicas para compreender com mais detalhes os perfis dos usuários e dos itens, o uso de informação contextual, a aplicação de avaliações dos usuários em diferentes critérios de um item, a coleta de avaliações não-explícitas para os itens, a flexibilidade no intuito de permitir que o usuário personalize suas recomendações e a aplicação de métricas mais adequadas para avaliar sistemas de recomendação (até então, a acurácia e a cobertura eram os métodos mais utilizados) · Duas publicações de 2010 ([JZFF10, LDGS11]) classificam sistemas de recomendação em seis tipos: demográficos (considera dados demográficos dos usuários), baseado em conhecimento (considera filtros de características previamente aplicados pelos usuários), baseado em comunidade (aplica viés nas recomendações conforme os relacionamentos detectados entre os usuários) além dos três tipos já citados, colaborativo, baseado em conteúdo e híbrido. Esses trabalhos também exploram questões latentes em sistemas de recomendação: confiança (níveis de relacionamentos entre usuários), explicações (detalhar para o usuário o motivo de ter recebido cada recomendação), persuasão (convencer usuários a consumir itens), recomendações para grupos de usuários e segurança (diferenciar avaliações reais de usuários de avaliações feitas com propósito de denegrir ou promover itens).

Conforme visto em [BOHG13], o número de publicações em congressos, conferências e *journals* na área de sistemas de recomendação quadruplicou de 2006 a 2012. Essa mudança possivelmente ocorreu devido ao aumento de dados gerados por diferentes aplicações relacionadas aos domínios citados no início desta seção, ao interesse da indústria e ao prêmio Netflix [BEL+07], que em 2006 ofereceu publicamente a quantia de um milhão de dólares para os autores do primeiro algoritmo que conseguisse melhorar em 10% a acurácia (medida por RMSE - *Root Mean Squared Error*) do sistema de recomendação atual da referida empresa. Os métodos de avaliação para sistemas de recomendação serão abordados na Seção 2.5.

2.2 Filtragem colaborativa

Nesta seção será abordada a filtragem colaborativa (FC), que é uma técnica para recomendações baseada no conhecimento coletivo, ou seja, baseia-se unicamente nas preferências dos usuários a respeito dos itens que compõe o catálogo do sistema. Serão verificadas quais são as formas tradicionais de implementar FC, quais são as vantagens e desvantagens desta abordagem bem como um exemplo prático. De acordo com Ekstrand *et al.* [ERK11], a FC pode ser utilizada entre usuários ou entre itens. Esses dois métodos de FC serão detalhados a seguir.

2.2.1 Colaboração entre usuários

A filtragem colaborativa entre usuários (FC-U) é uma técnica que envolve gerar previsões de avaliações para todos os itens e para todos os usuários de um sistema com base nas avaliações existentes. Ou seja, para saber se um item **x** será parte de uma recomendação para um usuário **a**, é preciso conhecer qual é a avaliação que os outros usuários do sistema fizeram para este item. A FC-U leva em consideração os usuários que têm uma preferência mais parecida com a do usuário a quem se deseja gerar recomendações. Para saber se os perfis de usuários são similares, algoritmos de cálculo de similaridade e distância podem ser utilizados. Na literatura é possível identificar diversos algoritmos para esta finalidade como a correlação de Pearson, distância Euclidiana, Jaccard, Minkowski, dentre outras. Conforme apresentado em Tan *et al.* [TSK05], medidas como a correlação de Pearson e distância Euclidiana são adequadas para dados densos enquanto medidas como Jaccard e Cosseno são adequadas para dados esparsos.

O resultado do cálculo de semelhança entre usuários pode ser armazenado em uma matriz de avaliações, onde cada avaliação pode ser um valor contínuo ou discreto, representando a preferência de cada usuário por cada item. É importante ressaltar que as avaliações podem ser explícitas (o usuário digitou ou selecionou a avaliação em uma escala) ou implícitas (o usuário comprou o item, consumiu o item por um tempo determinado, clicou no item, compartilhou o item em redes sociais, etc.). A forma de avaliação influencia na construção do algoritmo. Neste trabalho, serão consideradas as avaliações explícitas para a construção dos algoritmos.

Após montar a matriz de avaliações, as similaridades entre usuários podem ser calculadas através da aplicação de uma das medidas de similaridade e distância supracitadas. Para o problema de FC-U em sistemas de recomendação, a correlação de *Pearson* é a mais utilizada [BHK98]. A Equação 2.1 demonstra este cálculo. A notação utilizada é descrita a seguir: $s(u, v)$ representa a similaridade entre um usuário u e um usuário v, I_u representa o conjunto de itens avaliados pelo usuário u, $r_{u,i}$ é a avaliação que o usuário u fez para um item i, $\bar{r}_u$ representa a média de avaliações do usuários u. A correlação de *Pearson* reflete o relacionamento linear entre duas instâncias produzindo resultados no intervalo [-1, 1], sendo **0** a indicação de que não há relacionamento e **-1** ou **1** a indicação de que há uma correlação linear perfeita, direta ou inversamente proporcional.

$$s(u,v) = \frac{\sum_{i \in I_u \cap I_v}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I_u \cap I_v}(r_{v,i} - \bar{r}_v)^2}} \tag{2.1}$$

A seleção de usuários mais similares a partir da matriz de similaridades é geralmente feita através do k-NN (*k-Nearest Neighbors*), o algoritmo dos *k* vizinhos mais próximos. Segundo Ekstrand *et al.* [ERK11], o valor de k ideal varia de acordo com o domínio, sendo que para o domínio de filmes como o dataset MovieLens, por exemplo, um k entre 20 e 50 é adequado.

Os usuários mais similares são utilizados como base para calcular as preferências por itens para um determinado usuário. Após a realização destes cálculos, é possível ordenar de forma decres-

cente por preferência a lista de itens de um usuário e montar uma recomendação para este usuário (listando, por exemplo, os 10 itens com as maiores avaliações preditas). A fórmula para calcular avaliações está representada na Equação 2.2. Esta fórmula é utilizada para calcular uma avaliação $p_{u,i}$ de um usuário u para um item i com base nas avaliações feitas por todos os outros usuários (N). As similaridades entre usuários são utilizadas como pesos que são multiplicados pela diferença entre a avaliação de um usuário vizinho u' e a sua média $\bar{r}_{u'}$.

$$p_{u,i} = \bar{r}_u + \frac{\sum_{u' \in N} s(u,u')(r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} |s(u,u')|} \tag{2.2}$$

Um exemplo prático de cálculo de similaridades entre usuários para o domínio de filmes será demonstrado a seguir. Considerando a Tabela 2.1, pode-se identificar as avaliações de **5** usuários para **5** filmes, considerando uma escala entre **1 e 5**. O ponto de interrogação em uma célula representa que o usuário não avaliou o filme.

Tabela 2.1 – Avaliações de filmes

Filme	*O Regresso*	*Star Wars VII*	*Divertidamente*	*Mad Max Fury Road*	*Ponte dos Espiões*
Usuário A	2,0	?	?	4,0	2,0
Usuário B	5,0	5,0	4,0	5,0	?
Usuário C	?	4,0	5,0	5,0	3,0
Usuário D	4,0	4,0	3,0	?	1,0
Usuário E	?	5,0	4,5	5,0	1,5

Para prever a avaliação do **Usuário A** para o filme *"Star Wars VII"* deve-se primeiro calcular as avaliações médias dos usuários que avaliaram este filme, a similaridade entre o usuário e estes outros e, por fim, calcular a previsão da avaliação. As médias de avaliações dos Usuários **A** até **E** são, respectivamente: **2,66**, **4,75**, **4,25**, **3,00** e **4,00**. O cálculo da correção de Pearson para os usuários **A** e **B**, conforme a Equação 2.1 para este exemplo é:

$$s(A,\ B) = \frac{((2-2,66)*(5-4,75)) + ((4-2,66)*(5-4,75))}{\sqrt{((2-2,66)^2) + ((4-2,66)^2)} * \sqrt{((5-4,75)^2) + ((5-4,75)^2)}}$$

$$= \frac{-0,165 + 0,335}{1,49 * 0,35}$$

$$= \mathbf{0,32}$$

Consequentemente, as similaridades entre o Usuário **A** e os usuários **C**, **D** e **E** são iguais a: **0,84**, **0,32** e **0,74**. Após os cálculos de similaridades, a previsão de nota do Usuário **A** para o filme *"Star Wars VII"* pode então ser calculada, conforme a Equação 2.2.

$$p_{A,StarWarsVII} = 2,66 + \frac{(0,32 * (5 - 4,75)) + (0,84 * (4 - 4,25)) + (0,32 * (4 - 3)) + (0,74 * (5 - 4))}{|0,32| + |0,84| + |0,32| + |0,74|}$$

$$= 2,66 + \frac{0,93}{2,22}$$

$$= \mathbf{3,08}$$

A técnica de previsão de avaliações de usuários para itens é direta e simples de ser implementada. No entanto, é importante ressaltar que, pelo fato do algoritmo k-NN ser um *lazy learner*, ou seja, todos os dados de treino são utilizados no momento que deseja-se fazer uma classificação ou regressão, o tempo de execução aumenta à medida que aumenta o número de usuários e itens [TSK05]. A desvantagem em tempo de execução é compensada pela adaptabilidade do algoritmo, pois não é necessário gerar novos modelos à medida que os usuários interagem com o sistema, inserindo ou alterando suas avaliações. Também é importante observar que, como as avaliações estão em uma escala de 0,5 a 5,0 onde 0,5 é o pior (usuário não gostou) e 5,0 é o melhor (usuário gostou), pode-se considerar que dado o resultado da predição como 3,08, o Usuário A irá gostar do filme *Star Wars VII*, considerando um limiar igual a 3,0.

2.2.2 Colaboração entre itens

Conforme visto na Seção 2.2.1, a FC pode ser aplicada entre usuários. No entanto, essa abordagem é prejudicada pela escalabilidade do conjunto de dados à medida que o número de usuários cresce. Dependendo do domínio, o número de usuários pode ser muito maior que o número de itens. Além disso, a preferência dos itens é mais estável do que as preferências dos usuários [SKKR00]. Por ser mais estável, a computação de similaridades entre itens pode ser realizada *off-line* e um subconjunto contendo os itens mais similares a cada item pode ser mantido, o que pode representar um ganho de desempenho no momento de computar as recomendações.

As predições de avaliações feitas no modelo de colaboração entre itens (FC-I) podem ser feita de forma similar ao modelo FC-U. Primeiramente, para cada item a ser predito, é necessário identificar um conjunto S composto pelos itens mais similares a cada um destes itens avaliados por um usuário u. As similaridades entre os itens são calculadas utilizando uma medida como o cosseno dos vetores de cada item, sendo que um vetor é composto pelas avaliações atribuídas ao item. Para calcular a predição $p_{u,i}$ executa-se um somatório do produto da similaridade entre um par de itens i e j ($s(i,j)$) e a avaliação feita por um usuário u ao item j ($r_{u,j}$). Divide-se esse resultado pelo somatório do valor absoluto das similaridades entre os itens i e j [ERK11]. Esse cálculo está representado na Equação 2.3

$$p_{u,i} = \frac{\sum_{j \in S} s(i,j) r_{u,j}}{\sum_{j \in S} |s(i,j)|} \tag{2.3}$$

Assim como a FC-U, a FC-I pode ser implementada com uma variedade de algoritmos para computar similaridade ou distância. De acordo com Sarwar *et al.* [SKKR01], a similaridade do cosseno ajustado é a métrica mais indicada para FC-I, pois obteve melhores resultados de acurácia em comparação com a correlação de *Pearson* e cosseno puro. A fórmula do cosseno ajustado está representada na Equação 2.4, onde $\bar{r}_u$ representa a avaliação média do u-ésimo usuário, i e j são itens, U é o conjunto de usuários, $r_{u,i}$ e $r_{u,j}$ são as avaliações do usuário u aos itens i e j.

$$s(i,j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{i \in U} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in U} (r_{u,j} - \bar{r}_u)^2}} \tag{2.4}$$

A escolha por FC-I e FC-U é dependente do *dataset* utilizado, devendo-se considerar a proporção de itens versus a proporção de usuários. O tempo de execução do algoritmo kNN considerando a computação das similaridades para a FC-U é $O(|U|^2)$ e para a FC-I é $O(|I|^2)$, onde U e I representam o conjunto de usuário e o conjunto de itens, respectivamente.

Os algoritmos de FC-U e FC-I são fáceis de implementar e têm um bom desempenho. A FC-U oferece recomendações com maior chance de serendipidade, o que, no caso do *website* MovieLens, onde usuários avaliam filmes, resultou em uma maior satisfação [ERK11]. Serendipidade refere-se à surpreender o usuário de forma positiva. Técnicas avançadas como redução da dimensionalidade através de fatoração matricial podem ser empregadas para melhorar o desempenho das recomendações *online*.

2.3 Filtragem baseada em conteúdo

Diferente da FC que ignora os atributos dos itens para construir as recomendações, a filtragem baseada em conteúdo (FBC) é um modelo que tem como objetivo construir perfis para os usuários e para os itens tendo como base atributos dos itens. O processo de recomendação neste modelo consiste em contrastar o perfil do usuário com o perfil dos itens e recomendar a este usuário os itens com perfis mais similares aos dele ou ainda, contrastar o perfil dos itens que o usuário demonstrou preferência com cada item que deseja-se avaliar a preferência do usuário. Sendo assim, a FBC monta recomendações de itens semelhantes aos itens que o usuário demonstrou preferência no passado.

Para implementar um modelo de FBC é preciso primeiro definir quais serão os atributos utilizados para construir os perfis de itens e usuários. Esta abordagem para construção de perfil é denominada modelo de espaço vetorial (MEV). A partir das preferências dos usuários por cada item, pode-se determinar pesos para os atributos do MEV para, em seguida, realizar as análises de similaridades. Quando as fontes de atributos são textuais, os pesos podem ser definidos pelo algoritmo $TF\text{-}IDF$ (*Term Frequency-Inverse Document Frequency*), sendo que o objetivo deste

algoritmo é aumentar o peso de termos que ocorrem frequentemente em um documento (TF) e que ocorrem raramente no restante dos documentos (IDF). No domínio de filmes, os termos podem ser palavras-chave que melhor descrevem o filme, ou então, *tags* definidas pelos usuários [LDGS11].

Um exemplo de aplicação de FBC com TF-IDF é o trabalho de Mak *et al.* [MKP03] denominado INTIMATE, onde as sinopses de filmes são utilizadas para representar os vetores de perfil de usuários e itens. Os autores reportam que foi necessário realizar um pré-processamento nos textos a fim de aplicar técnicas de remoção de *stop-words*, que são palavras que ocorrem frequentemente no texto e não têm significado relevante, e *stemming*, que é a remoção de sufixos das palavras para gerar radicais.

Para que um sistema de recomendação possa aprender as preferências do usuário em um modelo FBC, pode-se usar algoritmos de classificação comuns na área de aprendizado de máquina. Exemplos desses algoritmos são Árvores de Decisão, Naïve Bayes e o próprio kNN, aplicação tradicional de FC [LDGS11].

De acordo com Burke [Bur02], as vantagens da FBC são:

- Independência do Usuário - as recomendações pode ser construídas sem depender da similaridade entre os usuários do sistema.
- Transparência - explicar recomendações geradas por FBC é simples, bastando indicar a descrição dos atributos que compõem o perfil dos itens.
- Novo item: o FBC é capaz de recomendar itens novos, com nenhuma avaliação de usuário.

Em contraposição, as desvantagens do FBC são ([Bur02, LDGS11]):

- Análise de conteúdo limitada - há uma dependência no número e tipo dos atributos utilizados para montar os perfis de itens e usuários. Mesmo uma seleção cuidadosa de atributos pode não oferecer uma representação boa o suficiente para distinguir itens.
- Superespecialização - por recomendar itens similares aos que o usuário demonstrou preferência, a FBC tende a não gerar recomendações com serendipidade, ou seja, que surpreendam o usuário.
- Novo usuário: é preciso haver uma determinada quantidade de avaliações de um usuário para que o sistema possa construir recomendações para o mesmo.

As desvantagens da FBC podem ser mitigadas através do uso de ontologias (limitação de análise de conteúdo) ou outros tipos de análise semântica, como os conceitos da Wikipedia. Já o problema da falta de serendipidade (superespecialização) pode ser resolvido inserindo recomendações randômicas com o auxílio de algoritmos genéticos, ou ainda, utilizando uma abordagem híbrida [LDGS11]. Sistemas de recomendação híbridos serão abordados na Seção 2.4.

Apesar de ser possível reduzir o impacto das desvantagens, a FBC é pouco utilizada na indústria [LDGS11]. Os autores comentam que uma provável causa deste fato é a criticidade das desvantagens do modelo para a maioria dos domínios. Outra causa pode ser pela dificuldade em extrair conteúdo dos itens. Mesmo que exista um grande número de atributos, é possível que esses atributos não sejam suficientes para representar a essência dos itens, ou seja, aquilo que os distingue, que faz com que os usuários o apreciem ou o rejeitem.

Trabalhos recentes foram publicados utilizando o modelo baseado em conteúdo com atributos extraídos de imagens e músicas. Exemplos desses trabalhos são [VdODS13], onde um modelo de fatores latentes para montar recomendações de músicas foi proposto e [YBL16], que explorou atributos de imagens para modelar perfis de usuários e inferir interesses em categorias determinadas como arte, carros e motocicletas, e jardinagem. Ambos os trabalhos utilizaram redes neurais convolucionais para a extração de atributos. Esse tema será detalhado no Capítulo 3.

Das duas formas tradicionais de recomendação, filtragem baseada em conteúdo e filtragem colaborativa, a segunda é mais utilizada tanto na indústria quanto na academia [NDRV09]. Um dos possíveis motivos para isto é a falta de conteúdo disponível para modelar perfis de usuários e itens. Porém, à medida que a Web foi se expandindo nos últimos anos, aumentando o acesso às redes sociais e o consumo de conteúdo multimídia por parte dos usuários, agora, não apenas texto está disponível para representar um item como também *tags*, áudio e vídeo. Além destas novas fontes de conteúdo, novas formas de explorá-los também foram criadas. Como por exemplo, chama a atenção o bom desempenho das redes neurais convolucionais para tarefas como classificação de imagens e detecção de objetos [KSH12, HZRS16].

Nas Seções 2.3.1 e 2.3.2, serão abordadas diferentes formas de representar conteúdo de itens criadas especificamente para o domínio de filmes e utilizadas como base de comparação para o método apresentado neste trabalho.

2.3.1 Abordagem baseada em representação textual

Dados como atores, diretor e escritor de um filme não representem o real conteúdo de um filme, mas sim, informação adicional sobre os mesmos [JZFF10]. Dessa forma, procura-se representar itens através de palavras-chave relevantes em vez de meta-atributos como os citados anteriormente. Uma forma ingênua de fazer essa representação é considerar todas as palavras que aparecem no texto de todos os itens e montar um vetor *booleano* onde apareceria o valor 1 para indicar a presença de determinada palavra e 0 para ausência. Essa abordagem tem como problema o fato de considerar que cada palavra relacionada ao item tem a mesma importância e, além disso, um sistema de recomendação com esses dados iria beneficiar documentos longos, com várias presenças do valor 1 nos vetores de tais itens [JZFF10].

Para resolver esse problema, pode-se utilizar uma técnica como a TF-IDF (*term frequency inverse document frequency*: frequência de um termo e inverso da frequência do documento). O TF-

IDF é uma técnica oriunda da área de recuperação de informações onde cada item (representado por um conjunto de palavras) é codificado como um vetor em um espaço Euclidiano. O TF-IDF é calculado através do produto da frequência de um determinado termo (TF) com o inverso da frequência do item (por exemplo: um documento) (IDF). A frequência do termo refere-se à quantidade de vezes que uma palavra aparece na descrição do item. Esta medida é calculada conforme a Equação 2.5, considerando $freq(i,j)$ o número de ocorrências de uma palavra i no item j e $totalPalavras(j)$ é o total de palavras associadas ao item. O IDF tem como objetivo reduzir o peso de palavras que aparecem muito frequentemente nos itens baseado no princípio de que palavras que se repetem muito são pouco importantes para distinguir um item. Seja N do número total de documentos e $N(i)$ o número de documentos de N onde a palavra i aparece, o IDF é então computado conforme a Equação 2.6. Finalmente, o TF-IDF de uma palavra i para um item j é computado conforme a Equação 2.7.

$$TF(i,j) = \frac{freq(i,j)}{totalPalavras(j)} \tag{2.5}$$

$$IDF(i) = \log \frac{N}{N(i)} \tag{2.6}$$

$$TF - IDF(i,j) = TF(i,j) \times IDF(i) \tag{2.7}$$

Para este trabalho, foram utilizadas duas formas de representar itens com conteúdo textual: as sinopses e as *tags* aplicadas por usuários aos filmes. A seguir, será apresentado um exemplo do cálculo TF-IDF para uma palavra. Esta representação foi inspirada no trabalho de Mak, Koprinska e Poon, [MKP03] e no trabalho de Cantador, Bellogín e Vallet [CBV10] que utilizaram TF-IDF para, respectivamente, representar conteúdo para sinopses e *tags* no contexto de recomendações.

Tabela 2.2 – Exemplo de filmes e suas sinopses

Filme	**Sinopse**
The Lord of The Rings: The fellowship of the ring (2001)	A meek Hobbit and eight companions set out on a journey to destroy the One Ring and the Dark Lord Sauron.
The Lord of The Rings: The two towers (2002)	While Frodo and Sam edge closer to Mordor with the help of the shifty Gollum, the divided fellowship makes a stand against Sauron's new ally, Saruman, and his hordes of Isengard.
The Lord of The Rings: The return of the king (2003)	Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.

De acordo com a Tabela 2.2, o TF da palavra ***Ring*** para o primeiro filme é o seguinte: $freq(Ring, 1) = 1$, $totalPalavras(1) = 21$, $TF = \frac{1}{21} \approx 0,048$. O IDF desta mesma palavra é o seguinte: $IDF = \log \frac{3}{2} \approx 0,176$. Sendo assim, o peso TF-IDF da palavra ***Ring*** para o primeiro item é $\approx 0,008$.

Tabela 2.3 – Exemplo de filmes e tags aplicadas aos mesmos

Filme	**Tags (quantidade)**
The Lord of The Rings: The fellowship of the ring (2001)	Fantasy (90), adventure (35), atmospheric (29), epic (14), music (13), scenic (9), stylized (6)
The Lord of The Rings: The two towers (2002)	Fantasy (76), book (25), magic (20), adventure (19), Action (12), war (11), wizards (9)
The Lord of The Rings: The return of the king (2003)	Fantasy (73), Oscar (32), adventure (31), magic (25), atmospheric (23), Tolkien (12)

Conforme a tabela 2.3 o TF da palavra **atmospheric** para o primeiro filme é o seguinte: $freq(atmospheric, 1) = 29$, $totalPalavras(1) = 196$, $TF = \frac{29}{196} \approx 0,148$. O IDF desta mesma palavra é o seguinte: $IDF = \log \frac{3}{2} \approx 0,176$. Sendo assim, o peso final TF-IDF para a palavra **atmospheric** é $\approx 0,026$.

2.3.2 Abordagem baseada em atributos de baixo nível

É importante observar que, para extrair dados sobre quadros de um filme, evitando fazer isso para todos os quadros do mesmo, pode-se utilizar apenas os quadros-chave de cada cena. Portanto, em um primeiro momento é preciso detectar as cenas em um vídeo. Uma abordagem padrão para fazer essa detecção é comparar cada quadro ao seu vizinho adjacente [DEC$^+$16, RSS05]. Uma similaridade intraquadros baixa geralmente indica uma mudança de cena, considerando-se um limiar pré-fixado. Tal similaridade é normalmente computada através de um histograma de intersecção $s(i)$ em um espaço de cores CSB (Cor, Saturação, Brilho). Em outras palavras, uma cena é definida como uma sequência de quadros obtidos por uma única câmera sem mudanças bruscas no conteúdo de cor das imagens consecutivas [RSS05]. A Figura 2.1 ilustra uma sequência de quadros de uma mesma cena.

Em tarefas de compreensão de vídeos, cada cena geralmente é representada por um único quadro denominado quadro-chave (*keyframe*), que é o quadro central da cena. A detecção de quadros-chave de cada cena é uma abordagem adotada tanto pelo método de baixo nível quanto pelo método baseado em redes neurais convolucionais, contribuição deste trabalho.

Uma abordagem de baixo nível pode utilizar estatísticas a respeito dos quadros-chave de um filme (trailer) com a finalidade de obter uma representação dos aspectos visuais artísticos de tais imagens. Tal abordagem, definida como extratora de conteúdo estético, foi descrita por Deldjoo *et al.* [DEC$^+$16] e será detalhada a seguir.

Foram selecionados os seguintes aspectos a serem extraídos dos quadros de trailers dos filmes: comprimento de cena, iluminação, variação de cor e movimento. Dessa forma, a representação final de um filme é um vetor $\mathbb{R}^{1\times5}$, dado por:

$$f_v = (\bar{L}_{sh}, \mu_{cv}, \bar{\mu_m}, \mu_{\sigma^2_m}, \mu_{lk}) \tag{2.8}$$

Figura 2.1 – Exemplo de uma sequência de quadros de uma mesma cena em um filme
Fonte: [IMG14]

onde $\bar{L}_{sh}$ é a média do comprimento de cada cena, μ_{cv} é a média da variação de cores, $\mu_{\bar{m}}$ e $\mu_{\sigma_m^2}$ são a média e desvio padrão do movimento acerca de todos os quadros e μ_{lk} é a média de iluminação sobre todos os quadros-chave.

Iluminação

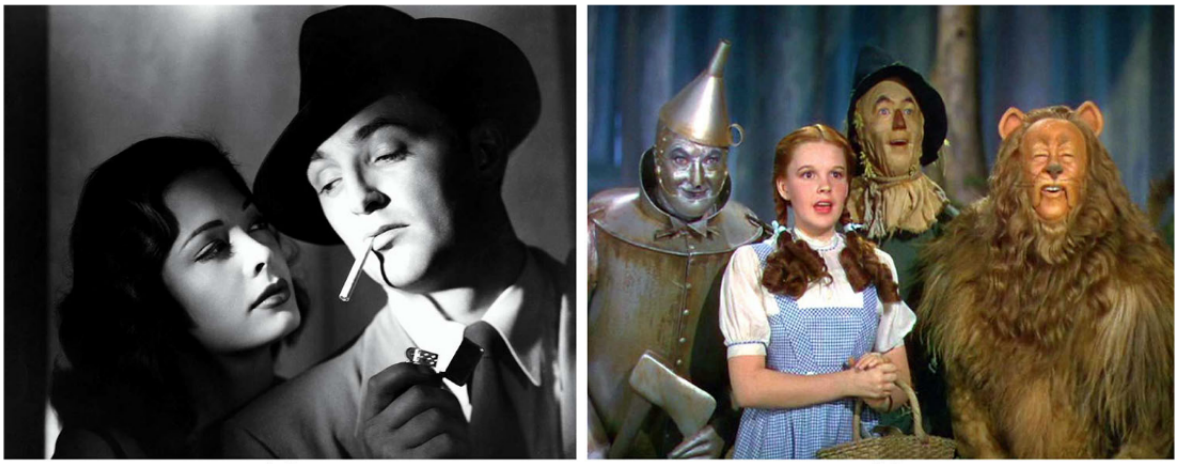
Há duas estratégias para iluminação em filmes: *chiaroscuro*, caracterizado por um alto nível de contraste entre áreas de sombra e áreas iluminadas, salientando as bordas dos objetos, e *flat*, que significa uma iluminação natural o mais próximo possível do real. A Figura 2.2 demonstra um exemplo de cada um desses tipos.

Filmes alegres como do gênero comédia geralmente possuem alta incidência de iluminação com pouco contraste entre as áreas claras e escuras enquanto filmes tensos como os do gênero horror e suspense possuem o efeito oposto. Os autores propuseram a captura da média e desvio padrão que corresponde ao brilho dos quadros.

Para calcular a iluminação de cada quadro, os autores fizeram uma transformação de todos os quadros-chave para o espaço de cores CSB e computaram a média μ e desvio padrão σ calculando a iluminação como: $\xi = \mu \cdot \sigma$.

Comprimento de cena

Outro atributo de baixo nível extraído por esta abordagem é o comprimento da cena. Esta medida é calculada como uma média, considerando o número total de quadros dividido pelo número de cenas detectadas. A intuição por trás desta medida é a de que filmes que contém ação terão



(a) **(b)**

Figura 2.2 – Exemplos de iluminação em quadros. **(a)** *Out of the past* (1947) representando alto contraste. **(b)** *The wizard of Oz* (1939) representando iluminação *flat*
Fonte: [DEC+16]

mais quadros devido ao movimento rápido da câmera e, como consequência disso, terão mais cenas, diferentemente de filmes dramáticos, que possuem cenas de conversação. A Equação 2.9 demonstra o cálculo de comprimento médio de uma cena.

$$\overline{L}sh = \frac{n_f}{n_{sh}} \tag{2.9}$$

onde n_f é o número total de quadros de um filme e n_{sh} o total de cenas.

Variação de cores

Segundo Deldjoo *et al.* [DEC+16], a variação de cor está fortemente associada ao gênero do filme. Como exemplo, diretores podem utilizar uma variedade de cores brilhantes para filmes de comédia e tons escuros para filmes de terror. Dessa forma, a determinante de uma matriz de variação de cores foi utilizada para medir a mesma. A Figura 2.3 demonstra dois exemplos de uso de tons de cores.

A variação de cor é calculada através da conversão dos quadros-chave em um espaço L*a*b* (L*=Luminosidade, a*=coordenada vermelho/verde, b*=coordenada amarelo/azul). Em seguida, uma matriz de covariância é gerada (Equação 2.10), e a variação de cor geral é obtida através da determinante desta matriz.

$$p_{cov} = \begin{bmatrix} \sigma_L^2 & \sigma_{Lu}^2 & \sigma_{Lv}^2 \\ \sigma_{Lu}^2 & \sigma_u^2 & \sigma_{uv}^2 \\ \sigma_{Lv}^2 & \sigma_{uv}^2 & \sigma_v^2 \end{bmatrix} \tag{2.10}$$

(a) (b)

Figura 2.3 – Exemplos de tons de cores em quadros. **(a)** *Django Unchained* (2012) representando predominância do tom vermelho para indicar violência. **(b)** *Lincoln* (2012) representando predominância do tom azul para indicar fadiga

Fonte: [DEC+16]

Movimento

O movimento em um vídeo é causado pelo movimento da câmera ou movimentos de objetos na cena. Enquanto que o movimento da câmera é capturado pela medida de comprimento da cena, o movimento dos objetos precisa de uma medida à parte. Este tipo de movimento é calculado com base na quantidade de *pixels* ativos dentro de um intervalo de tempo. Como o movimento é calculado sobre uma sequência de imagens, é necessário considerar todos os quadros de um vídeo. Em um quadro q, a média de movimento de *pixels* é representada por m_q e o desvio padrão do movimento de *pixels* é representado por $(\sigma_m)_t$. As Equações 2.11 e 2.12 demonstram o cálculo da média e desvio padrão de movimento em um quadro.

$$\mu_{\overline{m}} = \frac{\sum_{q=1}^{n_f} \overline{m}_q}{n_f} \tag{2.11}$$

$$\mu_{\sigma_m} = \frac{\sum_{q=1}^{n_f} (\sigma_m)_q}{n_f} \tag{2.12}$$

2.4 Sistemas de recomendação híbridos

Os filtros colaborativo e baseado em conteúdo são os modelos mais comuns encontrados na literatura e em artigos científicos. A definição de uma taxonomia precisa de todas as abordagens de sistemas de recomendação, no entanto, não existe. Em Ricci *et al.* [LDGS11], os autores citam os seguintes modelos: demográficos, baseado em conhecimento, baseado em comunidade e híbridos, além dos dois filtros comuns descritos anteriormente. Em Jannach *et al.* [JZFF10], os autores classificam sistemas de recomendações como sendo dos tipos: colaborativo, baseado em conteúdo, híbridos e baseado em conhecimento. O trabalho de Bobadilla *et al.* [BOHG13] indica apenas o modelo demográfico como alternativa a FC e FBC, descrevendo o modelo híbrido como uma

agregação dos três modelos. Por fim, o trabalho de Adomavicius e Tuzhilin [AT05] descreve que sistemas de recomendação podem ser resumidos em FBC, FC e híbridos. Este trabalho considera a classificação de sistemas de recomendação descritas naquele trabalho.

Sistemas de recomendação híbridos combinam um ou mais modelos de FC e FBC. Segundo [Bur02], esta abordagem pode ser dividida em categorias conforme descrição a seguir.

- **Baseado em Pesos**: neste modelo, o resultado da recomendação é obtido conforme escores atribuídos a cada técnica de recomendação presente no sistema. Como exemplo de aplicação há o trabalho de Claypool *et al.* [CGM+99], que propuseram uma média ponderada dos filtros colaborativo e baseado em conteúdo para um *website* de notícias. Através de experimentos, observou-se que a combinação entre os dois filtros reduziu a média de erros absolutos (*mean absolute error* - MAE) em comparação com cada filtro utilizado de forma isolada.

- **Alternado**: o sistema alterna entre FC e FBC de acordo com algum critério. Como exemplo, pode-se citar o trabalho de Billsus e Pazzani [BP00], onde o modelo alternado foi implementado utilizando como critério um grau de confiança de cada filtro, FC e FBC. Caso o FC não pudesse realizar uma recomendação, como no caso de haver um novo item, o modelo FBC assumiria. Essa mudança de abordagem durante a execução de um sistema de recomendação é denominada *fallback*.

- **Misto**: refere-se a misturar as recomendações geradas por FC e FBC. Apesar deste modelo evitar o problema de novo item, pois a FBC consegue realizar recomendações neste caso, não evita o problema de novo usuário. Em um sistema denominado PTV, Smyth e Cotter [SC00] utilizaram esta abordagem, com uma FBC abastecida com as descrições de programas de televisão (TV) e uma FC coletando informações sobre as preferências dos usuários do sistema. O objetivo desse sistema era personalizar as recomendações de shows de TV em uma agenda de horários para os usuários. Integrado a um sistema de TV digital, o sistema poderia também coletar dados como tempo que os usuários passam assistindo cada show para agregar aos perfis.

- **Combinação de Atributos**: ocorre quando se utiliza FC simplesmente como dados associados a cada item para complementar as tradicionais técnicas de FBC sobre os atributos de um *dataset*.

- **Cascata**: os dados gerados por um dos filtros, FB ou FBC, é utilizado como entrada para o outro. Um exemplo deste modelo híbrido é o sistema EntreeC, descrito em [Bur02]. Esse sistema utilizou conhecimento sobre restaurantes para fazer recomendações conforme os interesses dos usuários. Exemplos desse conhecimento são o preço, a localização e o tipo do ambiente dos restaurantes.

- **Aumento de Atributos**: uma técnica é utilizada para produzir uma avaliação de um item e este dado é passado a outra técnica de recomendação. Um exemplo desta abordagem é

o trabalho de [SKB+98] que propôs robôs de filtragem (*filterbots*) para aplicar avaliações a documentos utilizando critérios como a quantidade de erros ortográficos e o tamanho do texto. Um modelo FC poderia então ser aplicado utilizando humanos e robôs como usuários do sistema.

- **Meta-nível**: Diferentemente do modelo em cascata, na abordagem meta-nível o modelo de um FC é utilizado como entrada para um FBC, ou vice-versa. Por exemplo, no sistema proposto por Balabanović [Bal97], denominado Fab, os perfis dos usuários são modelados como um vetor de termos que descrevem as áreas de interesse dos mesmos. Agentes indexadores de páginas *web* são então utilizados para buscar documentos utilizando como base esses perfis.

A hibridização de FC e FBC pode ajudar a aliviar problemas comuns como o de novo item. No entanto, outros problemas, como novo usuário, ainda persistem. Conforme descrito em [JZFF10], os vencedores do prêmio Netflix utilizaram um modelo híbrido com a técnica baseada em pesos, sendo que os pesos foram determinados por uma análise de regressão linear e adaptados de acordo com os atributos dos itens e com o perfil dos usuários. Apesar de parecer a escolha certa para um sistema de recomendação, deve-se considerar questões como o desempenho e a complexidade de implementação, bem como realizar experimentos para avaliar os benefícios em domínios específicos.

2.5 Avaliação

A definição de métricas de avaliação é possivelmente tão importante quanto a seleção do modelo para implementação de um sistema de recomendação. Segundo Jannach *et al.* [JZFF10], alguns pesquisadores argumentam que os métodos de avaliação para sistemas de recomendação são limitados, especialmente aqueles que calculam o erro de predição de avaliação, pois existem muitos aspectos subjetivos para serem levados em consideração. Os autores afirmam ainda que, por esse motivo, é fundamental que haja evolução na definição de avaliações para que os modelos de recomendações possam ser comparados e empiricamente validados.

2.5.1 Acurácia de classificação

Um dos primeiros esforços para criar uma padronização de avaliação no contexto de sistemas de recomendação foi publicado em Sarwar *et al.* [SKKR00], onde dois métodos consagrados da área de recuperação de informações foram adaptados: ***precision*** e ***recall***.

Para medir *precision* e *recall* no contexto de sistemas de recomendação, considera-se o seguinte: seleciona-se um usuário para teste, oculta-se alguns dos itens que este usuário demonstrou preferência e executa-se as recomendações para prever um conjunto de itens que este usuário irá gostar. A partir dessas recomendações, quatro possibilidades de resultado são possíveis. Essas possibilidades estão representadas na Tabela 2.4. As métricas *precision* e *recall* refletem uma contagem

do número de itens que se encaixam em cada uma das possibilidades representadas na Tabela 2.4. Tais métricas estão definidas nas Equações 2.13 e 2.14 [SG11].

Tabela 2.4 – Classificação dos resultados possíveis de uma recomendação de um item para um usuário.

	Recomendado	Não Recomendado
Utilizado	Verdadeiro-Positivo (VP)	Falso-Negativo (FN)
Não Utilizado	Falso-Positivo (FP)	Verdadeiro-Negativo (VN)

É esperado um conflito entre *precision* e *recall*. Listas de recomendação longas tendem a melhorar o *recall* e a piorar *precision*. Em aplicações onde o número de recomendações apresentadas para o usuário não é pré-ordenado, é preferível avaliar os algoritmos em um intervalo de tamanhos de listas de recomendações, ao invés de usar um tamanho fixo. Dessa forma, é possível computar curvas contrastando *precision* e *recall*, ou ainda, a taxa de verdadeiros-positivos com a taxa de falsos positivos. As curvas do primeiro tipo são denominadas de curvas *precision-recall* e as curvas do segundo tipo são denominadas *Receiver Operating Characteristic* (característica de operação do receptor), ou curvas ROC. Curvas ROC enfatizam a proporção de itens que não são preferidos pelo usuário e que são recomendados, enquanto que as curvas de *precision-recall* enfatizam a proporção de itens recomendados que são preferidos pelo usuário. Uma medida que sumariza *precision* e *recall* é o $F1$-*score* ($F1$) [SG11]. A métrica $F1$ está representada na Equação 2.15. Como exemplo, considerando um conjunto de **10** itens que um usuário tem preferência, um resultado de uma recomendação que contém **3** desses itens em uma lista com **5** itens, *precision* é igual a **0,6**, *recall* é igual a **0,3** e F1 é igual a **0,4**.

$$precision = \frac{\#VP}{\#VP + \#FP} \tag{2.13}$$

$$recall = \frac{\#VP}{\#VP + \#FN} \tag{2.14}$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{2.15}$$

2.5.2 Acurácia de predição

Conforme especificado em [HKTR04], o erro absoluto médio (MAE) refere-se a obter a média aritmética dos erros de previsão das avaliações de usuários. Por exemplo, em um sistema que permite avaliações em uma escala entre **0** e **5**, um usuário que avaliou **3** itens com **3,7**, **4,5** e **4,6** e o sistema gerou predições iguais a **4,4**, **3,9** e **4,1** terá um **MAE** igual a **0,6**. Segundo os autores, o **MAE** é comumente implementado com variações denominadas *mean squared error* (erro quadrático médio - MSE), *root mean squared error* (raiz do erro quadrático médio - RMSE) e *normalized mean absolute error* (erro absoluto médio normalizado - NMAE). As variações **MSE** e **RMSE** incluem

uma elevação ao quadrado na diferença do valor predito subtraído da avaliação real, sendo que o resultado apresenta uma maior ênfase em erros maiores. A variação **NMAE** normaliza o resultado utilizando como parâmetro o intervalo de valores que o usuário faz as avaliações para que se possa fazer comparações em cenários e *datasets* diferentes. O **MAE** e suas variações estão demonstradas nas Equações 2.16 (a-d). Apesar de ser simples, direto e importante para ordenar itens que serão apresentados em uma recomendação, o **MAE** e suas variações podem ser pouco apropriados para a tarefa de encontrar bons itens, pois pode ser de pouca relevância ter uma boa acurácia em itens que o usuário não tem interesse [HKTR04].

$$MAE = \frac{\sum_{i=1}^{N} |predito - avaliado|}{N} \tag{2.16a}$$

$$MSE = \frac{\sum_{i=1}^{N} (predito - avaliado)^2}{N} \tag{2.16b}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (predito - avaliado)^2}{N}} \tag{2.16c}$$

$$NMAE = \frac{MAE}{Avaliacao_{max} - Avaliacao_{min}} \tag{2.16d}$$

2.5.3 Diversidade

A métrica de diversidade, também denominada de similaridade intralistas, propõe-se a verificar o quão diferentes são os itens de uma lista de recomendações. Quanto mais alto for o escore de diversidade, mais parecidos são os itens [ZMKL05]. É importante recordar que uma das desvantagens do modelo de recomendação baseada em conteúdo é a falta de diversidade, ou seja, a superespecialização. Dessa forma, uma métrica como a similaridade intra-listas pode sinalizar se uma solução para recomendações está conseguindo lidar bem com este problema. A Equação 2.17 demonstra o cálculo da similaridade intra-lista. Nesta equação, $recSet_u$ representa a lista de recomendações feitas a um usuário e $s(i,j)$ representa a similaridade de dois itens.

$$ILS_u = \frac{\sum_{i \in recSet_u} \sum_{j \in recSet_u, i \neq j} s(i,j)}{2} \tag{2.17}$$

De acordo com McLaughlin e Herlocker [MH04], medir a performance de um algoritmo utilizando *precision* e *recall* reflete melhor a experiência de um usuário do que medir a performance utilizando MAE porque, na maioria dos casos, o usuário recebe listas ordenadas de um sistema de recomendação ao invés de predições de avaliações em itens específicos. No trabalho, os autores determinaram que algoritmos que são bem-sucedidos pela métrica MAE produziram resultados insatisfatórios quando os itens posicionados na parte mais alta do ranking foram analisados.

Carenini e Scharma [CS04] argumentam que, sob um ponto de vista teórico, o MAE não é um bom indicador, pois todos os desvios recebem o mesmo peso. Por exemplo, um algoritmo prediz que um usuário irá avaliar um item como **3,5** em uma escala de **1-5**, e o usuário avalia este item com **2,5**. Esse erro é mais grave do que prever **1,5**, considerando que o usuário gosta de itens que são avaliados com nota maior ou igual a **3**. Ou seja, de acordo com a primeira predição, **3,5**, o usuário gostará do item, enquanto que, de acordo com a segunda predição, **1,5**, o usuário não gostará do item, o que de fato aconteceu. As duas predições têm um MAE igual a **1,0**.

As métricas de avaliação são de grande auxílio para a área de sistemas de recomendação, permitindo que algoritmos sejam comparados e que modelos robustos de experimentos sejam construídos. É importante utilizar mais do que apenas uma métrica para avaliar a qualidade de um sistema de recomendação para que esta avaliação seja o mais criteriosa possível.

3. REPRESENTAÇÃO NEURAL DE CONTEÚDO

Segundo Haykin [Hay08], redes neurais artificais (RNA), comumente denominadas apenas redes neurais, são processadores paralelos distribuídos constituídos de neurônios inspirados no funcionamento do cérebro humano. A função primordial de uma RNA é a classificação de instâncias com base em um modelo que é gerado a partir do treinamento da rede. Os neurônios de uma rede são dispostos em camadas, sendo que estes podem ser ativados ou desativados. Um neurônio computa um valor que é o resultado do produto de um vetor de parâmetros com os valores das entradas. Este valor é então passado a uma função de ativação, que vai determinar portanto, a ativação do neurônio. Uma arquitetura padrão simplificada de uma rede neural está demonstrada na Figura 3.1. Existem vários tipos de função de ativação, por exemplo: limiar, que ativa o neurônio quando o valor computado é maior ou igual a zero e sigmoide, onde o neurônio é ativado conforme o resultado de uma função sigmoidal como a logística ou a tangente hiperbólica, sendo que o neurônio é ativado quando o resultado desta função é maior que zero. De acordo com Haykin [Hay08], a função sigmoidal é a mais utilizada para RNA.

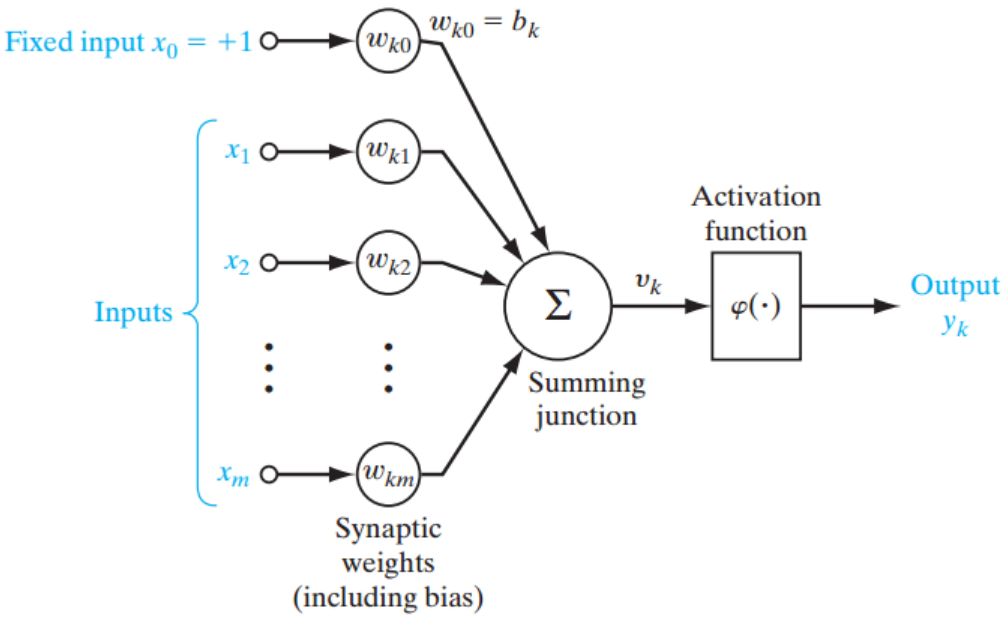


Figura 3.1 – Modelo de uma rede neural artificial
Fonte: [Hay08]

Conforme apresentado em Haykin [Hay08], a primeira camada de uma RNA é a entrada, sendo que os valores iniciais são os atributos de um *dataset* em conjunto com a unidade *Bias* que recebe um valor fixo igual a $+1$. Uma RNA pode ter diferentes arquiteturas, que define características como a forma como os neurônios estarão conectados e quantas camadas terá a rede. Exemplos de arquiteturas são a *Single-Layer Feedforward*, onde os neurônios da camada de entrada se conectam diretamente com a camada de saída, a *Multilayer Feedforward*, que tem a presença de mais de uma camada de neurônios, denominadas camadas escondidas, e as recorrentes, onde há ao menos um ciclo de retorno, ou seja, neurônios de uma camada posterior podem trocar dados com neurônios

de camadas anteriores. É imporante ressaltar que a diferença da camada de entrada para quaisquer outras camadas é que não há computação alguma nelas.

O processo de aprendizagem de uma RNA pode ocorrer de forma supervisionada ou não-supervisionada. O aprendizado supervisionado ocorre com um conjunto de exemplos onde a classificação de cada exemplo é conhecida. A rede é então treinada com o objetivo de ajustar os pesos para acertar a classificação do maior número possível de exemplos, sendo que este processo ocorre em iterações computando a soma ou a média dos erros de classificação. Estes erros normalmente são corrigidos através da aplicação do algoritmo *backpropagation*. O *backpropagation* corrige os pesos utilizando a regra da cadeia. O aprendizado não-supervisionado é utilizado quando se deseja encontrar algum tipo de padrão nos dados, sem o *feedback* de erros de classificação [Hay08].

Além da escolha de forma de aprendizado, as tarefas de aprendizado também devem ser determinadas para o uso de uma RNA. Diversas tarefas de aprendizado podem ser utilizadas, sendo que neste trabalho, será descrito apenas uma: o reconhecimento de padrões. O reconhecimento de padrões (classificação) é formalmente definido como o processo onde um padrão é atribuído para uma classe pré-definida. Uma RNA executa o reconhecimento de padrões através de uma sessão de treino sendo que as entradas são exemplos com identificação das suas respectivas classes. A rede treinada é então utilizada para classificar padrões ainda não vistos mas que pertencem a uma classe da população de exemplos utilizada para treino. A forma como a RNA executa a identificação de padrões é através da extração de informações dos dados de treino. O reconhecimento de padrões usando RNA pode ocorrer em uma das duas formas [Hay08].

O poder computacional de uma RNA está centrado na sua estrutura paralela massivamente distribuída e na sua habilidade em aprender. Essas capacidades fazem com que seja possível aproximar soluções para problemas complexos. Dentre as vantagens das redes neurais artificiais são citadas: não-linearidade, uma propriedade importante para o tratamento de entradas como sinais de som; adaptabilidade, referindo-se à capacidade de modificação dos pesos (parâmetros) de acordo com mudanças no ambiente; tolerância à falhas, no sentido de perder pouca qualidade se um neurônio está defeituoso; e resposta com evidências, o que significa que o nível de confiança necessário para, por exemplo, classificar um padrão, pode ser configurado [Hay08].

Em um dos primeiros trabalhos feitos para utilizar RNAs na tarefa de extração de atributos com redes que posteriormente vieram a ser denominadas convolucionais, LeCun *et al.* [LBD$^+$90] detalha uma rede com arquitetura multicamadas tendo como entradas imagens de tamanho 16x16 *pixels* de dígitos manuscritos. Foram utilizados técnicas que permitiram a identificação de atributos relevantes em camadas escondidas da rede denominadas *feature maps*. Várias camadas foram inseridas na rede, cada uma sendo responsável por extrair atributos diferentes da mesma imagem. Subsequente à extração de atributos, camadas de redução dimensional foram aplicadas. No total, a RNA foi constituída de 4 camadas — 2 para extração de atributos e 2 para redução dimensional, 4.635 neurônios e 98.442 conexões. A acurácia, a partir de um conjunto de treino composto de 7.291 imagens de dígitos manuscritos e 2.549 dígitos impressos, foi de **98,9%**.

Formalmente, uma rede neural convolucional (RNC) é um tipo especializado de RNA para o processamento de dados que possuem um formato de matriz, como séries temporais e imagens, que podem ser projetadas como matrizes de *pixels* em duas dimensões. O nome "convolução"indica que é feito o produto interno entre duas matrizes, sendo uma matriz o filtro e a outra matriz a entrada, em pelo menos uma das camadas da rede. Camadas de uma RNA tradicional utilizam multiplicação de matrizes de parâmetros entre os neurônios de duas camadas subsequentes. Isso significa que cada neurônio da camada posterior interage com todos os neurônios da camada anterior. Redes convolucionais, entretanto, possuem interações esparsas. Uma convolução é composta por três elementos: a entrada, que é uma matriz de valores numéricos, o *kernel*, que é uma matriz de dimensões menores do que a entrada, e a saída, que é o resultado da convolução. A saída é denominada *feature map*. Quando uma imagem é processada em uma RNC, a entrada pode ter milhões de *pixels*, sendo que, através de um *kernel* que ocupa dezenas de *pixels*, atributos significativos da imagem podem ser detectados [GBC16].

Uma camada típica de uma RNC consiste de 3 estágios: primeiro, várias convoluções são executadas em paralelo para produzir uma série de ativações lineares; em seguida, cada ativação é transformada através de uma função de ativação não-linear, estágio denominado detecção (*detector stage*); por fim, uma função de *pooling* é aplicada para modificar a saída da camada. Essa função substitui a saída por um resumo estatístico de saídas próximas. Por exemplo, a operação *max pooling* retira o maior valor de saída de uma matriz [GBC16]. Um exemplo da aplicação de uma convolução está representado na Figura 3.2. A Figura 3.3 demonstra o processo de transformação que uma imagem sofre ao receber várias convoluções.

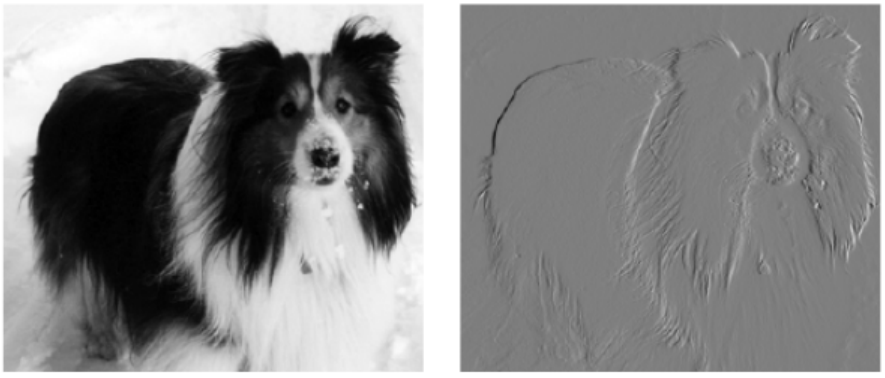


Figura 3.2 – Exemplo de aplicação de uma convolução
Fonte: [GBC16]

Os trabalhos de Krizhevsky *et al.* [KSH12] e Szegedy *et al.* [SLJ+15] detalharam RNAs mais profundas do que aquelas especificadas por LeCun *et al.* [LBD+90], ou seja, com mais camadas e neurônios, denominando-as com os termos hoje populares na área de aprendizado de máquina: redes neurais convolucionais (RNC) e *deep learning*. O primeiro trabalho propôs uma RNA com o propósito de classificar imagens do *dataset* ImageNet LSVRC-2010 composto por 1,2 milhões de imagens de 1.000 classes. A rede foi arquitetada com 5 camadas de convolução, cada uma

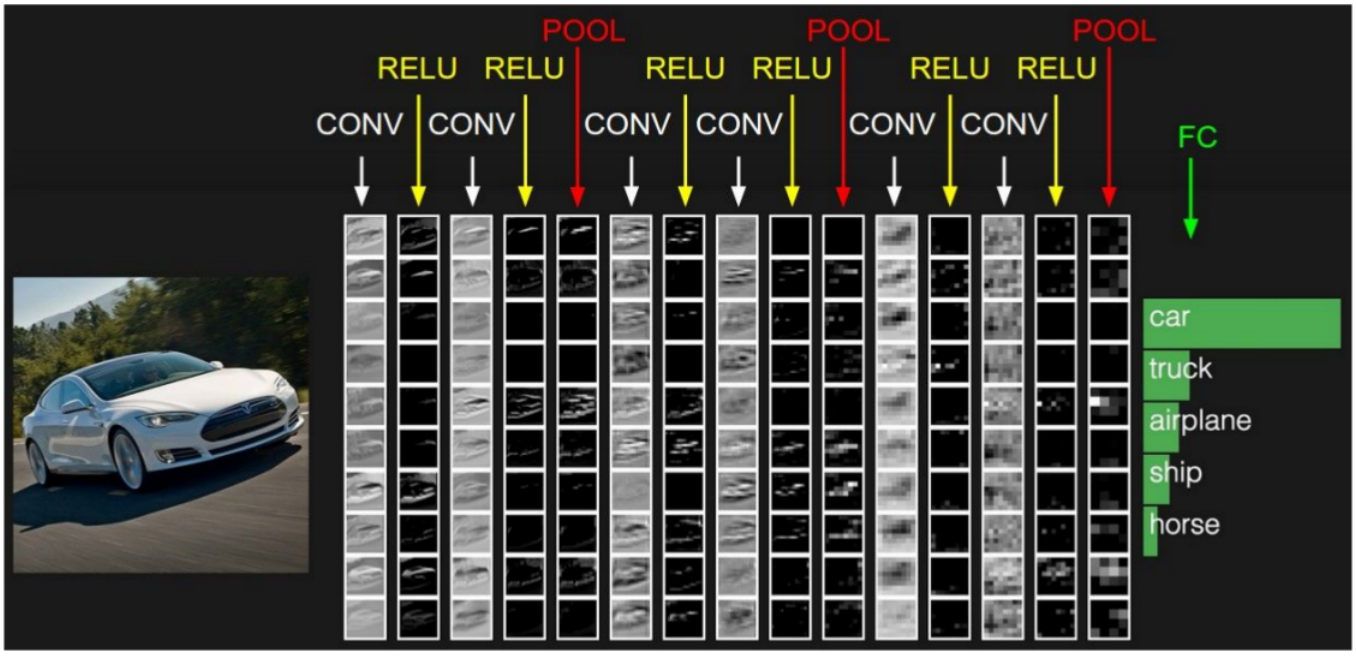


Figura 3.3 – Exemplo de aplicação de várias convoluções e pooling
Fonte: [Kar17]

seguida de uma camada de redução dimensional, 3 camadas totalmente conectadas e a camada de saída, totalizando 60 milhões de parâmetros e 650 mil neurônios. O resultado foi uma acurácia de **83%** (top-5). O segundo trabalho criou uma rede denominada *Inception* que, segundo os autores, foi responsável por definir o novo estado da arte para classificação e detecção no mesmo *dataset*, o ImageNet. A rede consiste em 22 camadas, envolvendo convoluções, reduções dimensionais e camadas totalmente conectadas. O resultado foi uma acurácia de **93,33%** (top-5).

A qualidade de reconhecimento de imagens e detecção de objetos aumentou drasticamente nos últimos anos graças ao avanço das redes neurais convolucionais para a realização dessas tarefas. A maior parte do progresso não é apenas devido à um hardware melhor, *datasets* e modelos maiores, mas uma consequência de novas ideias, algoritmos e arquiteturas das redes [SLJ+15].

Proposta por He *et al.* [HZRS16], a **ResNet** é uma RNC que tornou possível a aplicação de um número de camadas maior do que as redes convolucionais utilizadas até então. A ResNet venceu o desafio ILSVRC 2015 nas tarefas de classificação de imagens, localização e detecção de objetos [Lab15]. A entrada da ResNet é uma imagem de tamanho **224x224** *pixels* e a saída é uma camada totalmente conectada com 1.000 neurônios, devido ao fato de que o ImageNet contém 1.000 classes. A camada final é modificada conforme a quantidade de classes de um determinado conjunto de dados. Os autores testaram configurações com 18, 34, 50, 101 e 152 camadas, obtendo resultados melhores conforme o tamanho da rede aumentou. A ResNet-152 obteve, portanto, os melhores resultados. É importante observar que, para ocorrer a classificação de uma imagem, a mesma passa pela rede sendo transformada pelas camadas de convolução desde o princípio até a camada imediatamente anterior à camada final, sendo esta uma convolução que possui como saída um vetor de tamanho 1x2048 (para a ResNet nas versões contendo 50, 101 e 152 camadas). O

processo de convoluções aplicadas a uma imagem pode ser visualizado na Figura 3.4. Nesta figura, é possível notar que, em uma camada inicial de uma RNC, os filtros aplicados sobre a imagem original (convolução) resultam em atributos de baixo nível, imagens desfocadas e difíceis de interpretar. À medida que as convoluções vão sendo aplicadas na imagem, os padrões mais latentes ficam aparentes. No caso do automóvel da figura, é possível identificar itens como roda, espelho retrovisor e faróis, dependendo da parte (*crop*) que foi utilizada como entrada para a rede. Portanto, conclui-se que uma RNC pode ser utilizada como extratora de atributos de imagens.

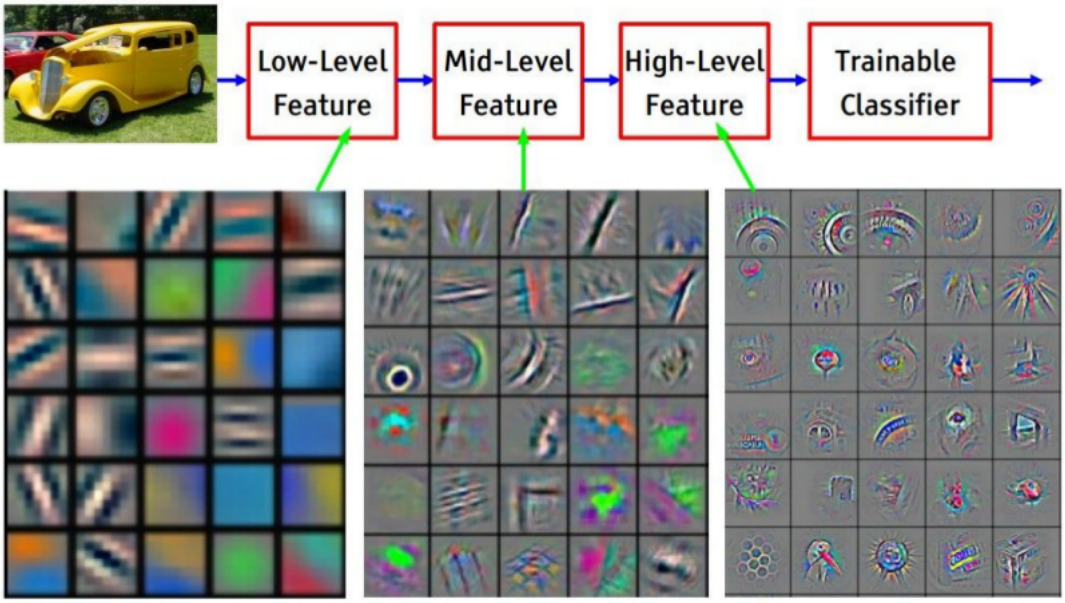


Figura 3.4 – Visualização da extração de atributos de uma imagem em uma RNC
Fonte: [LR13]

A abordagem baseada em RNA para este trabalho foi executada em duas etapas: extração de atributos e representação não-supervisionada, tendo sido baseada em atributos extraídos por uma RNC ResNet de 152 camadas. Considerando que filmes completos não estão disponíveis publicamente e também não são fáceis de processar devido ao alto custo computacional, decidiu-se utilizar trailers como fonte de informação visual da mesma forma que na abordagem de estética de mídia com atributos de baixo nível. De forma geral, trailers representam um resumo do roteiro de um filme, o que significa fornecer os atributos mais importantes no que diz respeito ao conteúdo do mesmo. O método baseado em redes neurais é um sistema automático que assiste os trailers dos filmes em uma base de dados e aprende as preferências dos usuários, para que posteriormente sejam geradas recomendações com base nas similaridades dos itens. Esse método é a principal contribuição deste trabalho.

Uma representação alto nível da arquitetura de um sistema de recomendação baseado em conteúdo inclui: i) um analisador de conteúdo, onde as representações dos itens são processadas e onde a técnica de extração de atributos é aplicada; ii) um aprendiz de perfis, onde o perfil de cada usuário é construído com base nos itens que o usuário indicou gostar e não gostar em seu histórico de ações no sistema; iii) um componente de filtragem, onde o perfil do usuário é explorado com

a finalidade de fornecer informações relevantes, como a geração de listas de itens ranqueados para recomendação [LDGS11].

Representação de Vídeo

Considerando o sucesso das RNCs para tarefas de compreensão de vídeos [SWBR16], decidiu-se investigar seu desempenho para a tarefa de recomendação de filmes. O processo utilizado para extrair os atributos dos trailers dos filmes é baseado em Simoes *et al.* [SWBR16] com as seguintes diferenças: i) a RNC utilizada é mais profunda, tratando-se de uma rede neural convolucional residual (ResNet) composta por 152 camadas; ii) a ResNet é pré-treinada em dois grandes conjuntos de dados, o ImageNet e o Places-365; iii) apenas os quadros-chave foram utilizados para a extração de atributos; iv) foi utilizado uma estratégia de seleção de 10 diferentes partes das imagens (10-*crop*) para obter vetores de atributos mais representativos.

O ImageNet é um dataset de imagens categorizado onde cada imagem pertence a uma dentre 1.000 classes disponíveis. O dataset possui mais de 14 milhões de imagens anotadas. Utilizando o dataset ImageNet, a competição ILSVRC (ImageNet Large Scale VIsual Recognition Challenge) foi criada em 2010 onde algoritmos são comparados nas tarefas de classificação de imagens, localização de objetos e detecção de objetos [RDS+15]. O ImageNet é, portanto, uma fonte rica de informações no que diz respeito a imagens. O ILSVRC fez surgir diversas arquiteturas de RNC, o que inclui a ResNet, criada por uma equipe de pesquisa da Microsoft em 2015.

O Places-365 é um dataset contendo 10 milhões de imagens rotuladas como categorias semânticas de cenas tendo uma grande abrangência sobre diversos tipos de ambientes do mundo e classificadas em 365 categorias [ZLX+14].

Diversos estudos fazem uso de RNCs pré-treinadas com o conjunto de dados ImageNet como extratores de atributos "de prateleira"[SRASC14]. Entretanto, o ImageNet é um conjunto de dados com foco em objetos, o que faz com que a RNC ignore outras informações contidas nas imagens. Entende-se que esses componentes que são ignorados, tais como lugares e ambientes, são importantes para a compreensão de conteúdo em trailers de filmes. Dessa forma, a RNC utilizada para este trabalho foi treinada sobre um conjunto composto de dados provenientes do ImageNet e do Places-365 para melhorar a habilidade da rede em reconhecer tanto objetos quanto ambientes ao mesmo tempo. Essa união resultou em cerca de 3 milhões de imagens (1,2 milhões do ImageNet e 1,8 milhões do Places-365) pertencentes a 1.365 classes (1000 do ImageNet e 365 do Places-365).

Para reduzir a complexidade computacional necessária para a execução deste método, decidiu-se extrair os atributos apenas dos quadros-chave dos trailers. A base de dados de trailers utilizada, LMTD [SWBR16], contém ≈ 30 milhões de quadros e ≈ 1 milhão de quadros-chave. Uma abordagem padrão para extração de atributos baseada em RNCs é utilizar uma parte central da imagem para representá-la. Entretanto, a maioria dos trailers estão no formato *wide-screen*, sendo que utilizar apenas uma parte central da imagem traz um risco de perder informações importantes da mesma. Dessa forma, decidiu-se utilizar uma estratégia de avaliação sobre 10 partes da imagem para fornecer vetores mais descritivos. Sobre os 10 vetores são calculadas as médias para gerar o vetor

final que pode ser considerado como a representação da cena, pois é extraído apenas um quadro por cena. A Figura 3.5 demonstra o procedimento de extração de atributos de cada quadro-chave.

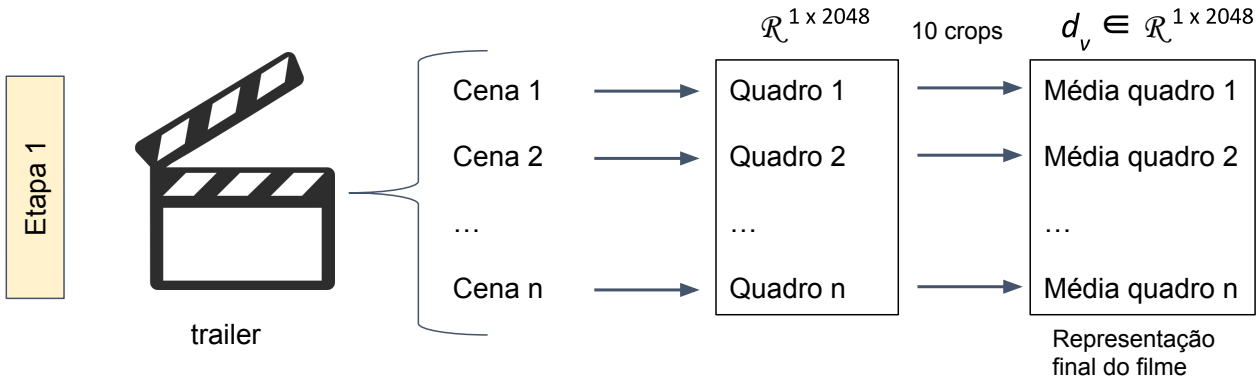


Figura 3.5 – Ilustração da primeira etapa de extração de atributos.

Para combinar as informações extraídas dos trailers, foi aplicado um algoritmo não-supervisionado utilizado para encontrar categorias de cenas, de forma similar à realizada em [SWBR16]. Seja $\mathcal{S}_j \in \{s_1, s_2, s_3, ..., s_n\}$ os atributos visuais baseados nas cenas do j-ésimo trailer que contém n cenas. As categorias das cenas foram encontradas através da aplicação do algoritmo *k*-means em uma amostra randômica de 100 mil vetores de atributos $s_i \in \mathbb{R}^{1\times2048}$. O algoritmo *k*-means foi executado 10 vezes para que uma inicialização apropriada pudesse ser encontrada. Este treinamento não-supervisionado fornece k centroides $c \in \mathbb{R}^{1\times2048}$ utilizados para construir um histograma semântico. Para construir esses histogramas, cada cena de um filme s_i é designada para o centróide mais próximo c_l no espaço Euclidiano. Desta forma, um trailer $\mathcal{T}_j$ é representado por um vetor de inteiros, dado por $\mathbf{h}_j \in \mathbb{I}^{1\times k}$, que é posteriormente normalizado a fim de realizar a transformação para a frequência relativa de cada grupo com $h_j = \frac{h_j}{\sum_i^k h_{ji}}$. Conforme reportado por Simões *et al.*, foram feitos testes de 2 a 150 grupos e percebeu-se que 128 grupos foi o valor mais adequado para a tarefa de classificação de gêneros dos filmes. Por este motivo, manteve-se este número de grupos neste trabalho. A Figura 3.6 ilustra este processo.

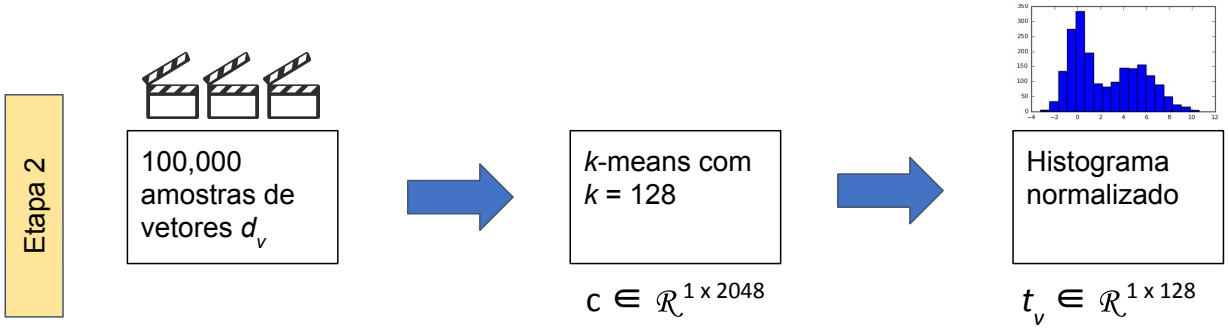


Figura 3.6 – Ilustração da segunda etapa de extração de atributos.

A Figura 3.7 representa exemplos de agrupamentos (*clusters*) encontrados através do algoritmo *k*-means. É possível notar que alguns *clusters* ficaram especializados em conceitos específicos (armas de fogo, explosões, faces) enquanto outros ficaram sensíveis à ambientes (planícies, lugares escuros, etc.). Exemplos do conteúdo semântico obtido por cada *cluster* são: i) o *cluster* 14 é especializado em cenas com armas de fogo, dentre outros conceitos; ii) o *cluster* 34 une quadros com elementos baseados em fogo; iii) o *cluster* 89 deu atenção a faces; iv) o *cluster* 99 agrupou cenas de início e fim de créditos; e v) o *cluster* 112 apresentou quadros com estilo visual típico de filmes de ficção científica. É importante destacar que todos estes *clusters* foram gerados automaticamente de forma não-supervisionada, ou seja, a RNC utilizada para extrair os atributos dos quadros não foi treinada para encontrar estes conceitos.

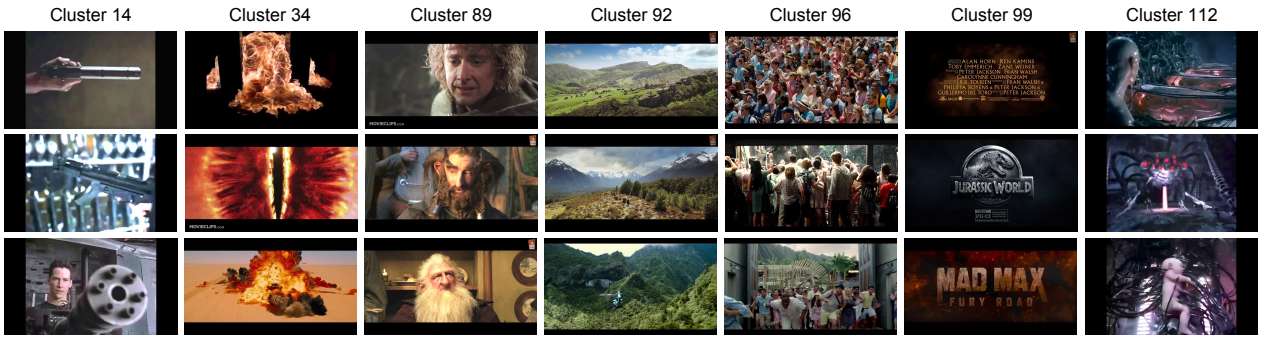


Figura 3.7 – Clusters de quadros-chave.

4. EXPERIMENTOS

Neste capítulo serão demonstrados a metodologia utilizada para montar o sistema de recomendação bem como os resultados obtidos através das métricas selecionadas.

O dataset utilizado para os experimentos é o resultado de uma intersecção entre o LMTD [SWBR16] que contém cerca de 3.500 trailers e foi utilizado como base para um trabalho de detecção de gêneros de filmes utilizando os quadros extraídos destes trailers, e o MoviLens versão "20M"[HK16] que contém 20 milhões de avaliações feitas por 138.000 usuários a 27.000 filmes. O dataset Movielens é mantido pela Universidade de Minnesota e sua versão "20M"é recomendada para pesquisas. A Tabela 4.1 demonstra as estatísticas do *dataset* utilizado para os experimentos.

Tabela 4.1 – *Dataset* utilizado para os experimentos.

Usuários	3.112
Filmes	3.473
Avaliações	316.971
Tags	664
Quadros-chave	416.760
± **Avaliações por Usuário**	101,85
± **Avaliações por Filme**	91,26
± **Tags por Filme**	202,02
± **Tags por Usuário**	124,46
± **Quadros-chave por filme**	120

Para agilizar o processo de recomendação as similaridades entre os itens e entre os usuários para cada um dos métodos foram pré-computadas e armazenadas como matrizes. As medidas utilizadas foram: para o método FC-U, a correlação de Pearson conforme definido na Equação 2.1, Seção 2.2.1; para FC-I, o cosseno ajustado, conforme definido na Equação 2.4, Seção 2.2.2; e para os demais métodos, foi utilizada a similaridade cosseno, conforme definido na Equação 4.5, Seção 4.1.

A partir dessas matrizes de similaridades foi feita uma análise manual dos itens mais similares para obter confiança nas medidas escolhidas. Portanto, foi escolhido um pequeno conjunto de filmes, a fim de encontrar os filmes mais similares aos mesmos dentro dos *datasets* LMTD [SWBR16] e MovieLens [HK16].

Por exemplo, foi escolhido o filme *The Matrix* (1999) e foram selecionados os filmes mais similares para cada abordagem baseada em conteúdo e FC-I. A expectativa, neste caso, foi de obter uma alta similaridade com os filmes *The Matrix Reloaded* (2003) e *The Matrix Revolutions* (2003), que formam uma trilogia. A Figura 4.1 demonstra o conteúdo recuperado. Cada linha da figura apresenta os 2 primeiros filmes mais similares a *The Matrix* seguido pelas posições que ficaram os outros 2 filmes da trilogia. A posição do ranking, que é ordenado por similaridade de forma decrescente, é apresentada no rodapé do pôster de cada filme. É importante lembrar que existem cerca de 3.500 filmes distintos na base de dados.

Figura 4.1 – Análise de similaridades entre filmes tendo como base o filme *Matrix* (1999) utilizando a medida cosseno. Primeira linha: resultados para a representação com atributos de baixo nível (estética de mídia), filmes: *Barely Lethal* (2015), *Redirected* (2014), *Matrix Revolutions* (2003) e *Matrix Reloaded* (2003). Segunda linha: resultados para a representação com atributos extraídos por RNC, filmes: *The Amityville Horror* (2005), *Death Race* (2008), *Matrix Revolutions* (2003) e *Matrix Reloaded* (2003). Terceira linha: resultados para representação textual com pesos TF-IDF sobre sinopses, filmes: *Hackers* (1995), *War Games* (1983), *Matrix Reloaded* (2003) e *Matrix Revolutions* (2003). Quarta linha: resultados para representação textual com pesos TF-IDF sobre *tags*, filmes: *Matrix Reloaded* (2003), *Matrix Revolutions*, *Tron* (2010) e *Blade Runner* (1982). Quinta linha: resultados para similaridades de avaliações entre itens, filmes: *The shawshank redemption* (1994), *Pulp Fiction* (1994), *Matrix Reloaded* (2003) e *Matrix Revolutions* (2003).

Nota-se que, utilizando *tags*, obteve-se alta similaridade entre os filmes da trilogia. Isso é esperado porque os usuários tendem a aplicar as mesmas *tags* para estes filmes. Outro detalhe a ser notado a respeito dessa trilogia é que o primeiro filme tem uma avaliação geral boa (média de 4,15 no Movielens) enquanto os outros dois filmes possuem médias gerais menores (3,33 para *Matrix Reloaded* e 3,18 para *Matrix Revolutions*). Possivelmente por esse motivo, a similaridade entre esses filmes na avaliação colaborativa (FC-I) é menor do que nas similaridades que consideram conteúdo. Na análise do método que utiliza RNC para extrair atributos, houve alta similaridade apenas com o segundo filme da trilogia, o que sugere que o trailer do terceiro filme é diferente dos outros dois filmes. Na análise de sinopses, o filme mais similar é o *Hackers* (1995), que possui as palavras *hacker* e *computer* na sua sinopse, da mesma forma que a sinopse do filme *The Matrix*.

4.1 Metodologia

Nesta seção a implementação do sistema de recomendação utilizado para este trabalho é descrita, sendo que os detalhes estão no Algoritmo 1. É importante observar que este algoritmo é utilizado tanto pelos métodos baseados em conteúdo quanto pelos métodos colaborativos. O que varia entre cada método é a medida de similaridade aplicada e a forma de realizar a predição da avaliação, que nos métodos colaborativos foram implementados de acordo com as equações demonstradas na Seção 2.2. O funcionamento do algoritmo é o seguinte: seja $\mathcal{U} = \{u_1, u_2, u_3, ..., u_n\}$ o conjunto de todos os usuários da base de dados. Para cada usuário $u_i \in \mathcal{U}$ um conjunto $\mathcal{R}_i = \{r_1, r_2, r_3, ..., r_m\}$ é gerado, onde m representa um total de 100 itens randômicos não avaliados pelo usuário u_i e todos os itens que este usuário avaliou com notas baixas ($< 3,0$). Além disso, para cada usuário $u_i \in \mathcal{U}$ um conjunto de teste $\mathcal{T}_i \in \{t_1, t_2, t_3, ..., t_n\}$ é criado, onde n representa todos os itens que o usuário u_i avaliou com notas altas ($> 4,0$). É importante destacar que o objetivo das métricas utilizadas para este trabalho é determinar se itens de preferência do usuário serão selecionados pelo sistema de recomendação.

A predição de avaliações é realizada com base em Ekstrand *et al.* [ERK11], para cada item em $\mathcal{R}_i$ e $\mathcal{T}_i$ e funciona da seguinte forma: a predição para um item i e um usuário u representada por $p_{u,i}$ (Equação 4.1) é igual ao somatório da avaliação feita pelo usuário u a um item j ($r_{u,j}$) contido em um conjunto I_u que representa todos os itens que o usuário u avaliou, subtraído da *baseline* do usuário-item $b_{u,i}$ e multiplicado pela similaridade entre o item j e o item i ($s(i,j)$) dividido pelo somatório do valor absoluto destas similaridades para, por fim, somar o resultado à baseline do usuário-item ($b_{u,i}$). A *baseline* de um usuário-item, $b_{u,i}$ (Equação 4.2), é obtida pela soma da média geral das avaliações, μ, com a *baseline* de um usuário, b_u, e a *baseline* do item, b_i. A *baseline* de um usuário, b_u, é igual ao somatório de cada avaliação feita pelo usuário u subtraído da média geral de avaliações, μ, dividido pelo total de itens que o usuário u avaliou. De forma similar, a *baseline* de um item i (Equação 4.4) é igual ao somatório de cada avaliação que o item i recebeu de um usuário u ($r_{u,i}$) subtraído da *baseline* de cada usuário u (b_u) que avaliou este item e da média geral de avaliações, μ, dividido pelo total de usuários que avaliou o item.

Algoritmo 1: Predição de avaliações e contagens de *hits*.

Data: $\mathcal{U}$: o conjunto completo de usuários.
Result: Total de hits: quantidade de itens do conjunto de teste contidos no subconjunto top-N.

```
begin
    for u_i ∈ U do
        hits ⟵ 0
        R_i ⟵ ObterFilmesRandomicos(u_i)
        T_i ⟵ ObterFilmesParaTeste(u_i)
        predTest ⟵ ObterPredicoes(T_i)
        predRandom ⟵ ObterPredicoes(R_i)
        predList ⟵ Concat(predTest, predRandom)
        predList.ordenarDescendente()
        while k ≤ N do
            if predList[k] ∈ T_i then
                hits ⟵ hits + 1
            k ⟵ k + 1
```

$$p_{u,i} = \frac{\sum_{j \in I_u} s(i,j)(r_{u,j} - b_{u,i})}{\sum_{j \in S} |s(i,j)|} + b_{u,i} \tag{4.1}$$

$$b_{u,i} = \mu + b_u + b_i \tag{4.2}$$

$$b_u = \frac{1}{|I_u|} \sum_{i \in I_u} (r_{u,i} - \mu) \tag{4.3}$$

$$b_i = \frac{1}{|U_i|} \sum_{u \in U_i} (r_{u,i} - b_u - \mu) \tag{4.4}$$

Tabela 4.2 – Notação para as Equações.

Variáveis	Descrição
$b_{u,i}$	Baseline de avaliações para o usuário u, item i
μ	Média de todas as avaliações
b_u	Baseline de avaliação do usuário
b_i	Baseline de avaliação do item
I_u	Itens avaliados pelo usuário u
$r_{u,i}$	Avaliação do item i atribuída pelo usuário u
U_i	Usuários que avaliaram o item i
$p_{u,i}$	Predição de avaliação para o usuário u, item i
$s(i,j)$	Similaridade cosseno entre os itens i e j
f	Vetor de atributos

Após realizadas as predições, ambos os conjuntos, $\mathcal{R}_i$ e $\mathcal{T}_i$, são unidos e ranqueados em ordem decrescente pela avaliação predita, sendo $\mathcal{F}_i$ o conjunto resultante dessa combinação. Por fim, toda vez que um item que pertence ao conjunto $\mathcal{T}_i$ aparece em um subconjunto top-N de $\mathcal{F}_i$, um *hit* é computado. Nesse contexto N refere-se à quantidade de itens com as maiores notas

preditas contidas no conjunto de itens $\mathcal{F}_i$. A Figura 4.2 representa a primeira e a segunda etapa, respectivamente, do método utilizado para implementar e avaliar o sistema de recomendação para cada uma das abordagens utilizadas (colaborativas e baseadas em conteúdo).

O conjunto de avaliações dos usuários a ser considerado na Equação 4.1 poderia ser limitado a uma vizinhança que consideraria apenas os itens mais similares, conforme [ERK11]. Entretanto, através dos experimentos foi possível notar que utilizar apenas similaridades positivas no numerador da equação levou a resultados melhores. A métrica cosseno foi utilizada para calcular as similaridades entre os itens (Equações 4.5 e 4.6) para as abordagens baseadas em conteúdo. A Tabela 4.2 define a notação para essas equações.

Conforme descrito no início deste capítulo, para a abordagem colaborativa entre usuários (FC-U) utilizou-se a como medida de similaridade a correlação de Pearson descrito na Equação 2.1 e para a abordagem colaborativa entre itens (FC-I) utilizou-se como medida de similaridade o cosseno ajustado descrito na Equação 2.4. A vizinhança considerada para ambos os métodos FC-I e FC-U foi igual a 50 (usuários para FC-U e itens para FC-I) [ERK11].

$$s(i,j) = \frac{f_i \cdot f_j}{\|f_i\|_2 \, \|f_j\|_2} \tag{4.5}$$

$$||f||_2 = \sqrt{{f_1}^2 + {f_2}^2 + ... + {f_n}^2} \tag{4.6}$$

É importante lembrar que se definiu que um item precisa ter uma avaliação $> 4,0$ para ser incluído no conjunto de teste $\mathcal{T}_i$. No conjunto de dados utilizado as avaliações encontram-se no intervalo [0,5, 5,0]. O tamanho do conjunto de filmes randômicos, $\mathcal{R}_i$, foi fixado em 100 e o tamanho do subconjunto top-N variou de 1 a 15, aumentando em 1 a cada iteração.

A Equação 4.2 representa uma *baseline* para a predição de notas: considera-se a média sobre todas as notas do sistema somada a *baseline* para cada usuário, b_u, e a *baseline* para cada item, b_i. As Equações 4.3 e 4.4 descrevem, respectivamente, as *baselines* de usuários e itens.

O trabalho de Cremonesi *et al.* [CKT10] foi utilizado como inspiração para avaliar o desempenho do sistema de recomendação. Métricas como o erro absoluto médio (MAE), raiz quadrada do erro quadrático médio (RMSE) e erro absoluto médio normalizado (NMAE) têm sido bastante utilizadas para sistemas de recomendação [BOHG13]. Essas métricas são úteis para avaliar a qualidade da predição de notas, tendo como desvantagem o fato de considerar os erros de forma igualitária para todos os itens. Considerando que itens com pouca relevância para um usuário têm baixa probabilidade de impactar o desempenho de um sistema de recomendação, outras métricas foram selecionadas. Para avaliar o aspecto de acurácia de classificação das listas de recomendação, utilizou-se a métrica $F1$ que consolida precisão (*precision*) e revocação (*recall*). No contexto de sistemas de recomendação, *recall* mede a razão entre itens relevantes selecionados (verdadeiros positivos) e todos os outros itens que são relevantes mas não foram selecionados (falsos negativos). A métrica *precision* mede a razão entre os verdadeiros positivos e todos os outros itens selecionados (falsos

positivos). Portanto, *precision* representa a probabilidade de um item selecionado ser relevante, enquanto *recall* representa a probabilidade de um item relevante ser selecionado [HKTR04].

Para analisar com mais detalhes a qualidade do sistema de recomendação proposto, um métrica adicional foi selecionada, a similaridade intra-listas [ZMKL05] demonstrada na Equação 2.17. Esta métrica é utilizada para medir a diversidade de uma lista de recomendações. A diversidade pode aumentar a chance de recomendações com características de novidade e serendipidade. Uma recomendação com característica de novidade leva em consideração as preferências do usuário. Por exemplo, um filme estrelado por um ator ou atriz que o usuário gosta pode ser considerado uma novidade, tendo grande chance de agradar o usuário. Uma recomendação com serendipidade refere-se à surpreender positivamente o usuário com um item que é diferente do seu padrão de preferências. Serendipidade, por definição, é também uma novidade.

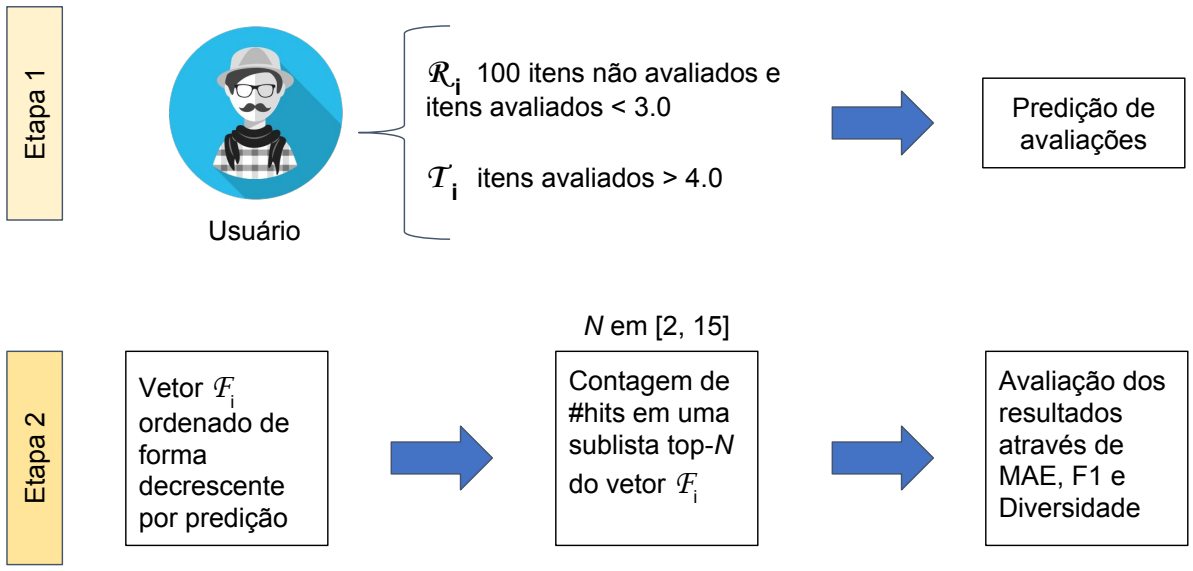


Figura 4.2 – Etapas 1 e 2 do sistema de recomendação.

4.2 Resultados

Nesta seção serão demonstrados os resultados obtidos através das métricas selecionadas para avaliar a qualidade de um sistema de recomendação construído com as representações visuais, textuais e colaborativas. O objetivo da aplicação dos experimentos é validar se o método introduzido por este trabalho, que monta uma representação de conteúdo através da aplicação de uma rede neural convolucional é superior aos demais métodos de representação de conteúdo e competitivo com os tradicionais métodos de recomendação baseados em filtragem colaborativa.

Os métodos implementados e avaliados estão descritos a seguir. É importante observar que cada método tem como objetivo fazer a predição de avaliação para cada item, conforme descrito nas

Seções 2.2.1, 2.2.2 e 4.1. Fora avaliados quatro métodos baseados em conteúdo (Baixo Nível, RNC, Tags e Sinopse), dois métodos baseados em colaboração (FC-U e FC-I) e, por fim, uma hibridização que faz a média aritmética entre cada par possível entre os métodos.

- **Baixo Nível**: a representação de cada item é um vetor de 5 posições cujo propósito é capturar a estética da mídia conforme descrito na Seção 2.3.2.
- **RNC**: a representação de cada item é um histograma normalizado conforme a classificação de cada quadro-chave do *trailer* correspondente ao filme conforme descrito no Capítulo 3. Este método é introduzido neste trabalho.
- **Tags**: a representação de cada item é um vetor com pesos determinados pela técnica $TF\text{-}IDF$ sobre cada *tag* aplicada a um filme, conforme descrito na Seção 2.3.1.
- **Sinopse**: a representação de cada item é semelhante ao método de **Tags** descrito acima, com a diferença de que os pesos $TF\text{-}IDF$ são calculados sobre cada palavra contida na Sinopse de cada filme, conforme descrito na Seção 2.3.1.
- **FC-U**: neste método não há representação do item de maneira que as recomendações são construídas de acordo com as similaridades entre usuários, conforme descrito na Seção 2.2.1.
- **FC-I**: neste método cada item é representado por um vetor de avaliações, conforme descrito na Seção 2.2.2.
- **Híbrido**: a predição da avaliação deste método é uma combinação entre cada par possível de métodos, exemplos: Baixo Nível + RNC, RNC + FC-I, etc. Como foi definido o peso de 50% para cada método, uma simples média aritmética foi aplicada para obter a predição de avaliação para os itens. O total de combinações possíveis entre os pares dos 6 métodos é igual a 15, porém, para simplificar a representação gráfica, foram selecionados as hibridizações que obtiveram os melhores resultados para cada combinação. Neste caso, cada método combinado com FC-U obteve os melhores resultados, reduzindo a quantidade de híbridos de 15 para 5.

4.2.1 Comparação entre métodos baseados em conteúdo

Esta seção demonstra os resultados obtidos das métricas selecionadas para os métodos baseados em conteúdo. A questão a ser respondida é a de que o método apresentado por este trabalho, RNC, é superior aos métodos Baixo Nível, Tags e Sinopse. A primeira métrica aplicada foi a MAE (Erro Absoluto Médio). Como cada método é responsável por prever uma avaliação para cada usuário e para cada item (filme), é natural medir a qualidade desta predição a partir do erro absoluto médio, conforme definido na Equação 2.16(a) descrito na Seção 2.5.2. Esta medida representa o valor médio dos erros de predição de avaliação considerando todos os usuários do conjunto de dados. Para esta métrica nota-se que o método RNC obteve o melhor resultado,

superando o segundo colocado, Tags, em 7% e o terceiro colocado, Baixo Nível, em 14%. O pior resultado foi obtido com o método Sinopse. A Tabela 4.3 demonstra os resultados.

Tabela 4.3 – Erro absoluto médio por método baseado em conteúdo em ordem crescente.

Posição	**Método**	**MAE**
1	RNC	**0,9104**
2	Tags	0,9749
3	Baixo Nível	1,0415
4	Sinopse	1,1970

A segunda métrica aplicada para comparar os métodos baseados em conteúdo foi a $F1$. Novamente, deseja-se verificar se o método RNC é superior aos concorrentes. É importante observar que, para obter esta métrica, foram utilizadas 14 iterações, variando o tamanho da lista de recomendação de 2 a 15 (representado por N). A Figura 4.3 ilustra os resultados. O gráfico plota o $F1$ obtido por cada método em cada iteração (N, no eixo "x"). É possível perceber que, nas três primeiras iterações a diferença entre o método RNC e o outro método de representação de conteúdo visual, Baixo Nível, foi pequena. No entanto, a partir da iteração que utilizou listas de recomendação com 5 itens a diferença passou a ser bastante significativa, sendo ampliada constantemente até a última iteração, onde a diferença entre o método RNC e o segundo colocado foi de cerca de 0,15 pontos o que indica que o método RNC possui média harmônica entre *precision* e *recall* superiores. A diferença entre o método RNC e os métodos de representação textual foi ainda maior, ficando em cerca de 0,3 pontos de diferença para o método de representação por Tags e em cerca de 0,5 pontos de diferença para o método de representação por Sinopses.

A terceira métrica aplicada para comparar os métodos baseados em conteúdo foi a diversidade. Esta métrica tem como objetivo indicar o quão diferente são os itens dentro de uma lista de recomendação. A intuição por trás de uma lista diversa é a de que há uma maior chance de surpreender o usuário com recomendações que não são óbvias. O objetivo em utilizar esta métrica neste trabalho é verificar se métodos de recomendação baseados em conteúdo geram listas de recomendação com baixa diversidade (problema da super-especialização). Os resultados desta medida demonstram que, apesar do método que utiliza RNC gerar listas mais diversas que o método baseado em atributos de baixo nível, o mesmo foi derrotado pelos demais métodos baseados em conteúdo, Sinopses e Tags. Uma possível interpretação para isto é que as sinopses dos filmes, por exemplo, mudam bastante, tendo pouca semelhança mesmo com filmes que compõe uma trilogia. O problema da falta de diversidade é conhecido por afetar sistemas de recomendação baseados em conteúdo conforme descrito na Seção 2.5.3. É importante lembrar que, quanto **menor** o escore, maior é a diversidade da lista de recomendação. Apesar do método RNC ter sido derrotado, a diferença de diversidade para os métodos de representação textual Tags e Sinopse foi pequena, ficando mais evidente a partir da décima iteração. A Figura 4.4 ilustra os resultados. O gráfico indica a diversidade obtida por cada método em função da quantidade de itens nas listas de recomendação (N) variando de 2 a 15.

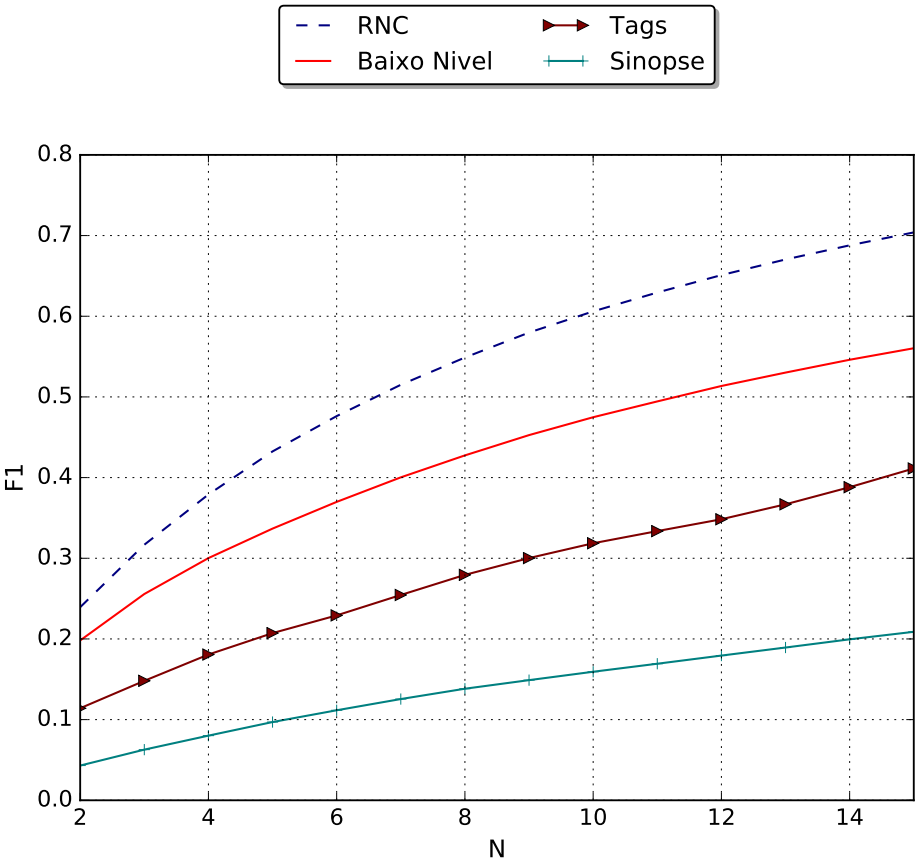


Figura 4.3 – $F1$ para os métodos baseados em conteúdo.

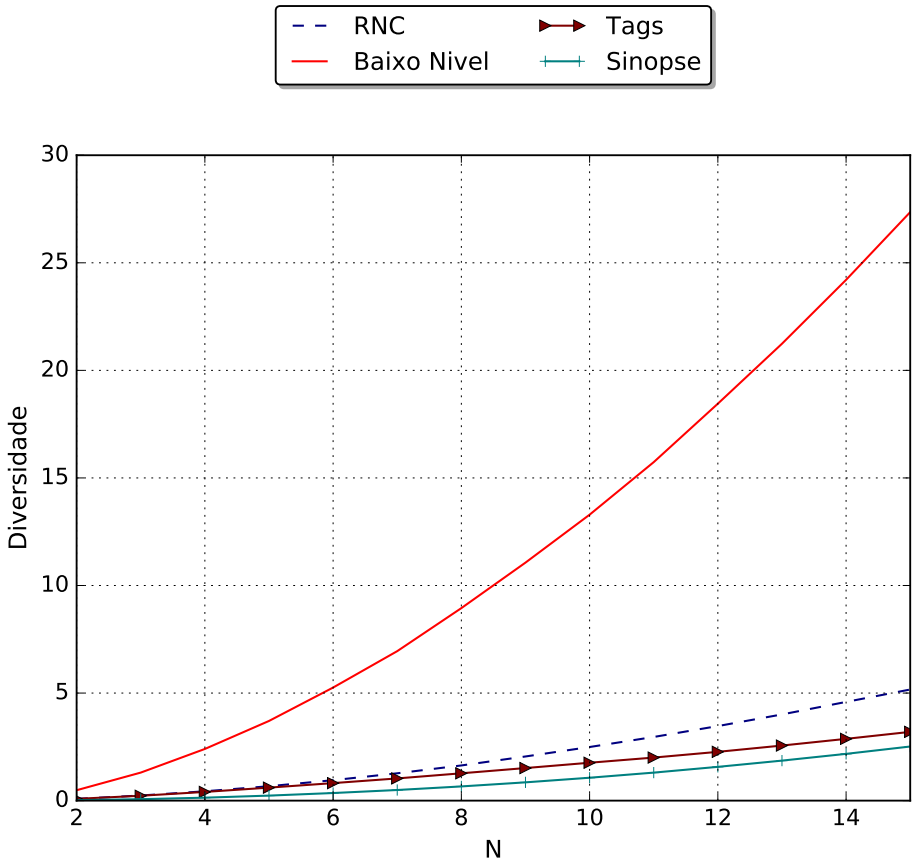


Figura 4.4 – Diversidade para os métodos baseados em conteúdo.

4.2.2 Comparação do método RNC com métodos colaborativos

Esta seção compara o método baseado em conteúdo proposto neste trabalho, RNC, com os tradicionais métodos baseados em colaboração, entre usuários (FC-U) e entre itens (FC-I). Foram aplicadas as três métricas utilizadas na comparação entre os métodos baseados em conteúdo: MAE, $F1$ e diversidade. O objetivo é verificar se o método RNC é competitivo com os métodos colaborativos. Para a primeira métrica, MAE, descrita na Tabela 4.4, o melhor resultado foi do método FC-U seguido por FC-I, o que indica que os métodos colaborativos geram predições de avaliações mais precisas que o método RNC.

A segunda métrica, $F1$, ilustrada na Figura 4.5, indica uma vitória do método FC-U sobre o método RNC a partir da terceira iteração ampliando a vantagem até a última iteração. No entanto, é importante observar que a diferença entre FC-U e RNC foi pequena e, além disso, o método RNC superou o método de colaboração entre itens, FC-I, em todas as iterações, obtendo uma vantagem significativa. Estes resultados indicam que o método RNC é promissor para o contexto de recomendações.

A terceira métrica, diversidade, também foi utilizada para comparar o método RNC com os métodos colaborativos. Os resultados desta métrica, ilustrados na Figura 4.6, indicam que os métodos colaborativos geram listas de recomendação mais diversas que o método RNC em todas as iterações, sendo que a diferença fica mais evidente a partir da sétima iteração. No entanto, é importante lembrar que o problema da baixa diversidade é conhecido para os sistemas de recomendação baseados em conteúdo e que a diferença entre os métodos colaborativos para o método RNC para listas com até 8 itens foi pequena.

Tabela 4.4 – Erro absoluto médio por método em ordem crescente.

Posição	**Método**	**MAE**
1	FC-U	**0,7446**
2	FC-I	0,8318
3	RNC	0,9104

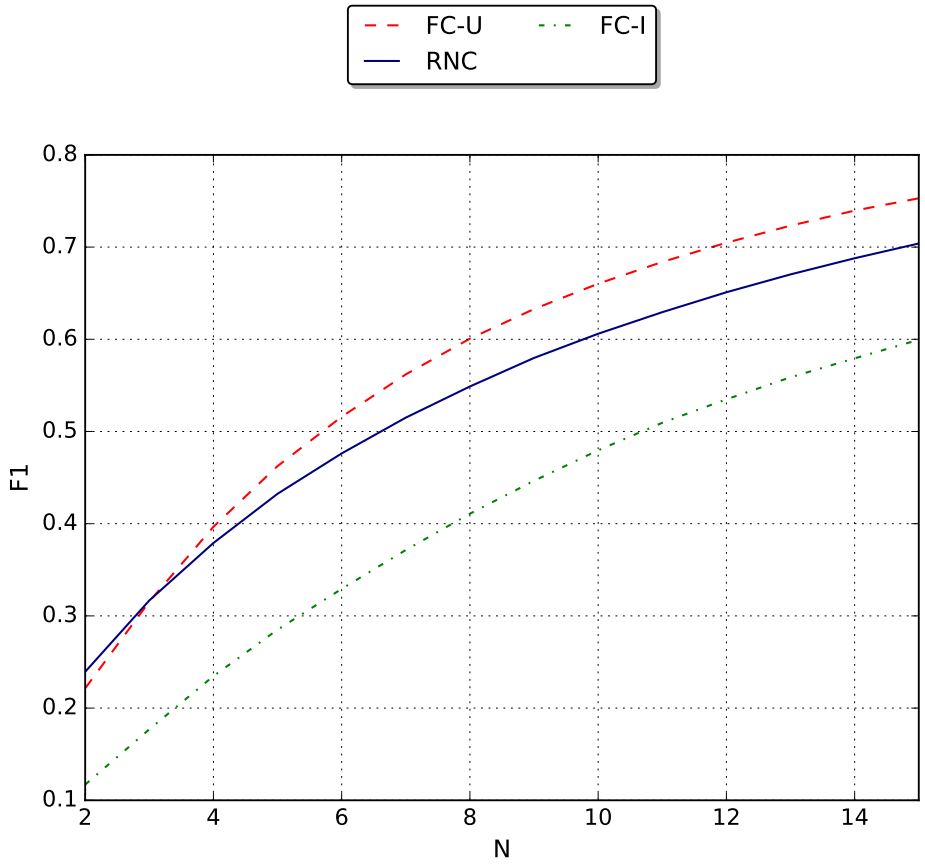


Figura 4.5 – F1 para os métodos colaborativos e o método RNC.

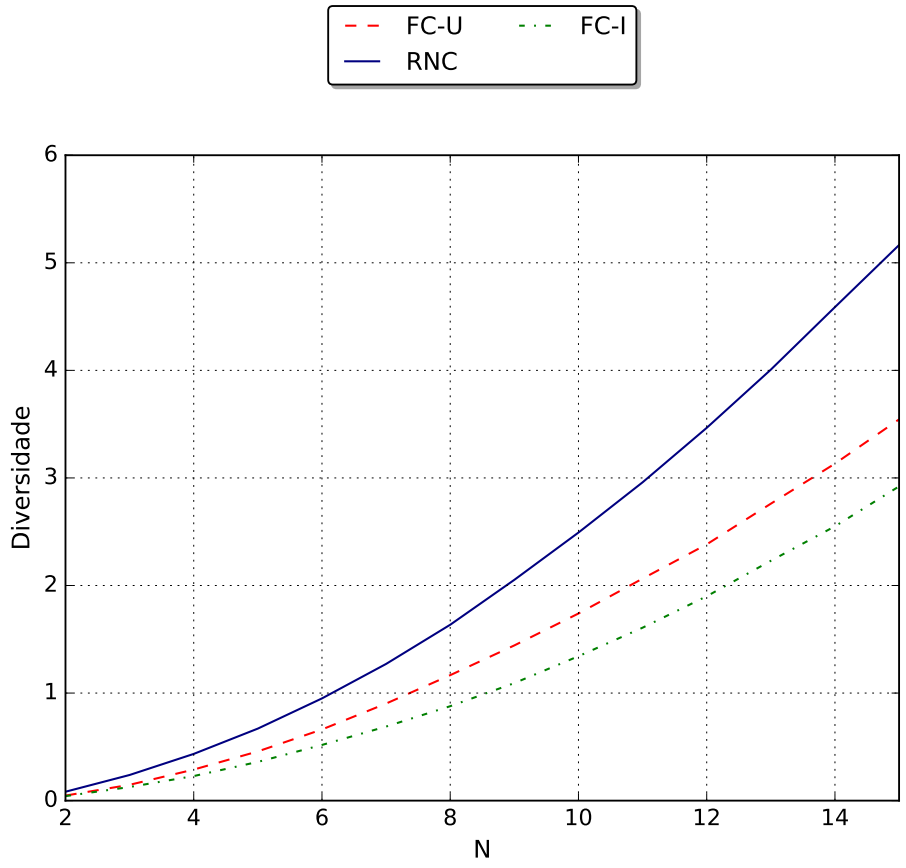


Figura 4.6 – Diversidade para os métodos colaborativos e o método RNC.

4.2.3 Comparação entre hibridizações

Nesta seção compara-se as médias aritméticas obtidas entre as combinações de pares de métodos. Conforme descrito no início deste capítulo, considerou-se apenas as hibridizações que obtiveram os melhores resultados. Por exemplo, o método Sinopse obteve os melhores resultados quando combinado com FC-U. Novamente foram aplicados as três métricas: MAE, $F1$ e diversidade. Para a primeira métrica, MAE é possível notar que o melhor resultado ainda é do método FC-U utilizado individualmente. No entanto, foi possível melhorar os resultados para todos os métodos baseados em conteúdo quando combinados com FC-U com destaque para o método apresentado neste trabalho, RNC. Hibridizado com FC-U, o método RNC obteve o terceiro melhor resultado de precisão em predição de avaliações tendo uma diferença bastante pequena em comparação com FC-U e com a hibridização FC-U + FC-I.

Tabela 4.5 – Erro absoluto médio por combinação de métodos em ordem crescente. Inlcui os métodos individuais FC-U, FC-I e RNC.

Posição	**Método**	**MAE**
1	FC-U	**0,7446**
2	FC-I + FC-U	0,7518
3	RNC + FC-U	0,7873
4	Tags + FC-U	0,8021
5	FC-I	0,8318
6	Baixo Nível + FC-U	0,8475
7	RNC	0,9104
8	Sinopse + FC-U	0,9235

A segunda métrica, $F1$, ilustrada na Figura 4.7, indica que as hibridizações geram resultados bastante parecidos em todas as iterações. No entanto, é importante destacar que a combinação do método RNC com o método FC-U gerou os melhores resultados.

Por fim, a terceira métrica, diversidade, ilustrada na Figura 4.8, foi utilizada para comparar as combinações entre os métodos. Para essa métrica, de forma similar ocorrida com $F1$, nota-se que os resultados foram bastante parecidos, sendo que a hibridização entre Sinopse+FC-U gerou as listas de recomendação com maior diversidade. Novamente é importante lembrar que as sinopses dos filmes variam bastante, o que provavelmente influenciou esses resultados.

4.2.4 Tempo de Execução

O tempo de execução é uma medida relevante para ser considerada pois, para ser usado em produção, o algoritmo deve ter um bom tempo de resposta. Nessa perspectiva, os métodos baseados em conteúdo obtiveram os melhores resultados, sendo o método RNC, que obteve tempo de execução médio, por usuário, $\approx 0,0291$ o menor dentre todos. Portanto nota-se que, apesar do método FC-U ter obtido melhores resultados em F1, diversidade e MAE, o método RNC possui um tempo

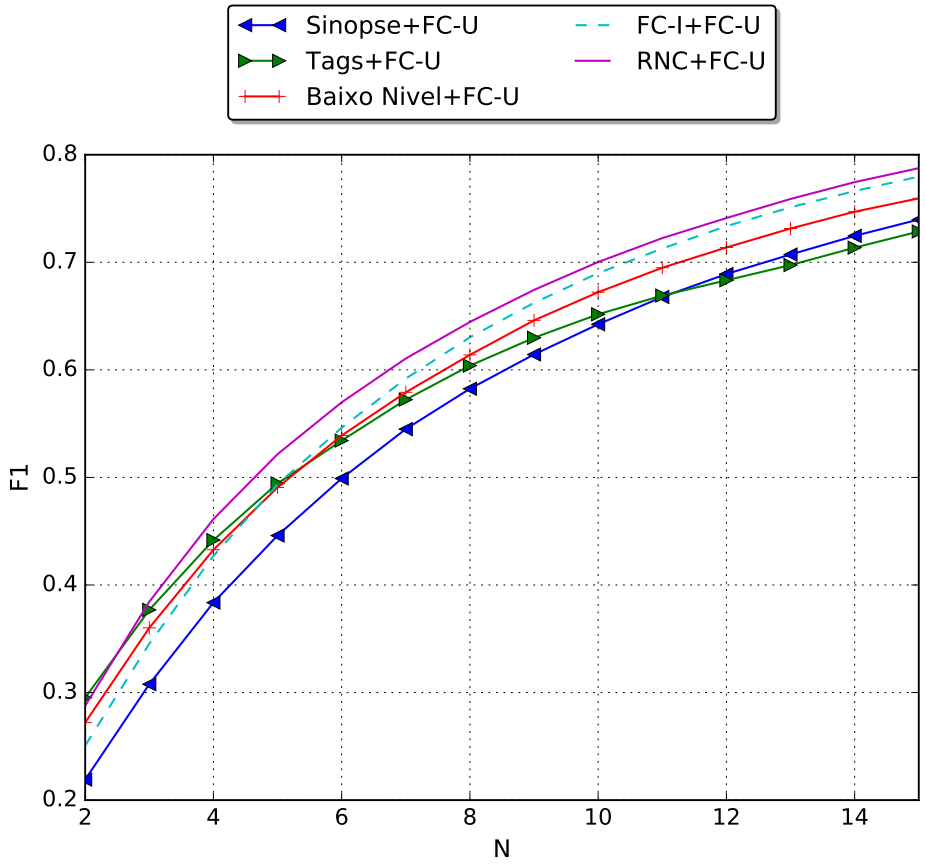


Figura 4.7 – F1 para os métodos híbridos.

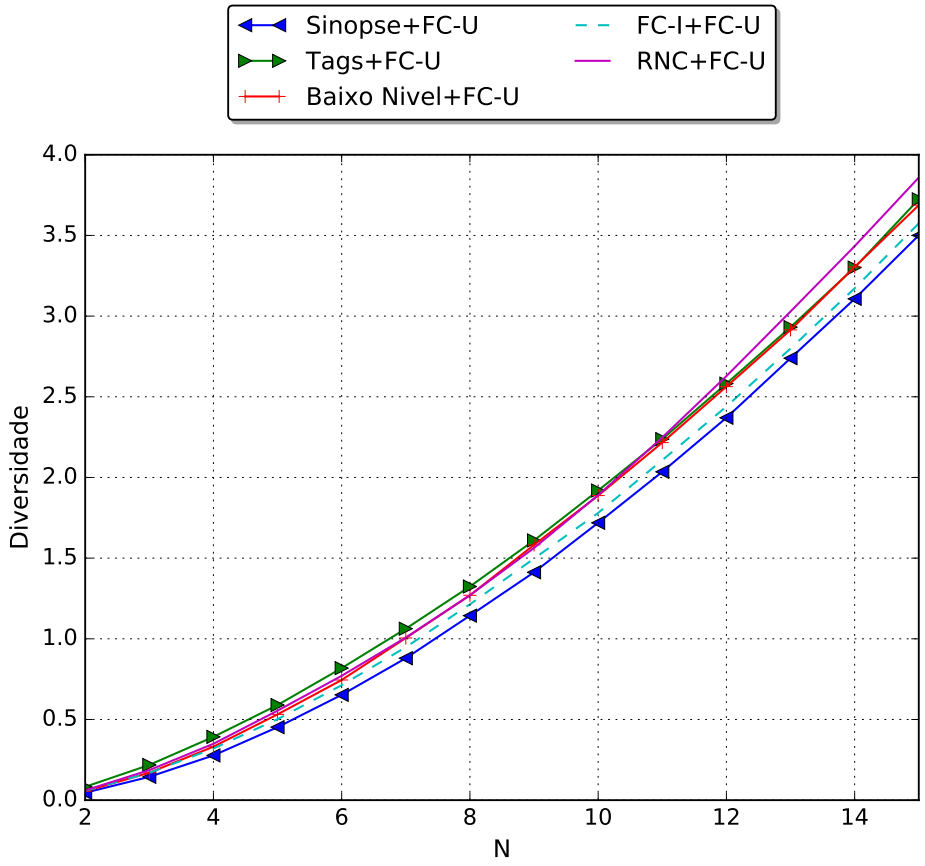


Figura 4.8 – Diversidade para os métodos híbridos.

de execução significativamente menor. A Tabela 4.6 ilustra os resultados. É importante lembrar que, para agilizar os cálculos de predição de avaliação executados para gerar as recomendações, as similaridades entre os itens foram pré-computadas. O tempo de execução demonstrado na Tabela 4.6 demonstra, portanto, apenas a execução dos cálculos de predição de avaliações para cada método.

Tabela 4.6 – Tempo de execução, por usuário, para cada método de recomendação.

Método	**Tempo de execução médio (segundos)**	**Tempo de execução total (minutos)**
RNC	**0,0291**	**1,5093**
Baixo Nível	0,0293	1,5196
Tags	0,0616	3,1949
Sinopse	0,0632	3,2779
FC-U	6,7514	350,1726
FC-I	51,2841	2659,9353

5. CONCLUSÕES

Sistemas de recomendação são soluções para indicação de itens para usuários de acordo com suas preferências. Dos dois tipos mais comuns de sistemas de recomendação, a filtragem colaborativa e a filtragem baseada em conteúdo, a primeira é a que tem mais trabalhos publicados em um período recente bem como é a solução mais utilizada pela indústria. Os motivos para este fato são: a tarefa de extração de atributos significativos de itens é complexa, as fontes de dados são escassas e recomendações baseadas em conteúdo tendem a gerar sugestões muito parecidas com as que o usuário já conhece, desta forma fazendo com que haja pouca utilidade na recomendação.

Este trabalho apresenta o uso de redes neurais convolucionais como extratora de atributos visuais de trailers de filmes com o objetivo de verificar se o sucesso destas redes na detecção de objetos em imagens bem como na classificação de imagens também ocorre no contexto de recomendações. Especificamente, apresentou-se um método que assiste cada um dos trailers associados a um filme da base de dados, detecta os quadros-chave de cada uma das cenas destes trailers, obtém a média dos atributos extraídos de 10 partes de cada um destes quadros-chave para, posteriormente, construir um histograma de classificação de quadros obtido através de um aprendizado não-supervisionado que consiste na classificação de cada um dos quadros dos trailers. Os histogramas são utilizados como forma de representação dos itens e servem como base para a aplicação de uma equação que tem como objetivo gerar a predição de avaliações que os usuários atribuirão aos filmes. Para isto, foi feito uma união dos conjuntos de dados de trailers LMTD [SWBR16] com o Movielens [HK16], que contém dados de avaliações de usuários aos filmes.

Como *baseline* de comparação com o método apresentado foi utilizado um método de detecção de estética de mídia que avalia aspectos como iluminação, movimento, variação de cores e comprimento de cenas coletados sobre os trailers dos filmes, e também dois métodos de análise textual que atribuem pesos através da técnica $TF\text{-}IDF$ para as palavras contidas ora nas sinopses dos filmes, ora como forma de *tags*.

Através das métricas de erro absoluto médio, $F1$ e diversidade, foi possível demonstrar que o método que utiliza rede neural convolucional teve um bom desempenho, superando todos os outros métodos baseados em conteúdo em termos de qualidade de listas de recomendação e acurácia na predição de avaliações, o que confirmou que as novas fontes de conteúdo são capazes de melhorar o desempenho dos sistemas de recomendação.

Em comparação com os tradicionais métodos de recomendação baseados em colaboração entre usuários e itens FC-U e FC-I, o método que utiliza RNC mostrou-se bastante competitivo tendo sido superado por FC-U nas métricas aplicadas por uma diferença pequena. Este fato deixa uma oportunidade para melhoria para o método aqui apresentado. Como tentativa de melhorar os resultados construiu-se métodos híbridos, através da combinação das notas preditas entre pares de métodos. Destes híbridos, o método com RNC e a filtragem colaborativa entre usuários, FC-U, produziu o melhor resultado em termos de qualidade das listas de recomendação ($F1$) e o terceiro melhor resultado em termos de acurácia de classificação (MAE).

Também é importante ressaltar que, apesar de ter sido derrotado em métricas tradicionais de sistemas de recomendação, o tempo de execução médio por usuário dos métodos baseados em conteúdo (0,2 segundos em média) é menor que os métodos colaborativos (6,8 segundos para filtragem colaborativa entre usuários e 36 segundos para a filtragem colaborativa entre itens) com destaque para o método que utiliza RNC.

Outro ponto importante é que os métodos colaborativos precisam ser re-computados a cada período determinado de tempo pois os itens recebem novas avaliações dos usuários e, além disso, o usuários podem alterar avaliações já feitas. Esta re-computação não precisa ser tão frequente para a abordagem baseada em conteúdo, já que as fontes de dados como o trailer do filme não muda, bastando aplicar a extração de atributos para cada novo filme que é adicionado ao catálogo.

5.1 Limitações e Trabalhos Futuros

Uma limitação do método apresentado neste trabalho é o uso dos trailers para representar filmes. Apesar de haver a intuição de que o trailer é uma representação precisa e compacta de um filme, nem todo trailer pode representar este fato. Outro motivo que levou a utilizar trailers ao invés do filme completo é que os trailers estão disponíveis publicamente. Também é importante considerar que, o tempo que demora para extrair os atributos de cada trailer foi cerca de 2 minutos. Como foram utilizados 3.473 filmes, o tempo total de extração de atributos com a rede neural convolucional foi cerca de 113 horas, quase 5 dias. Cada trailer possui um tempo de 2 a 3 minutos.

Como oportunidade de melhoria sugere-se extrair atributos do áudio dos trailers bem como explorar outras formas de combinação que utilize as informações textuais obtidas das sinopses das tags e dos atributos como diretor, produtor e atores dos filmes o que podem levar a uma representação mais rica de conteúdo e, possivelmente, melhorar o desempenho das recomendações.

5.2 Publicação

O método que faz uso de uma rede neural convolucional para representar conteúdo de itens (RNC) em comparação com o método que analisa estética de mídias (Baixo Nível) foi publicado na International Joint Conference on Neural Networks no ano de 2017 sob o título "*Leveraging deep visual features for content-based movie recommender systems*".

REFERÊNCIAS BIBLIOGRÁFICAS

[AT05] Adomavicius, G.; Tuzhilin, A. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", *IEEE Transactions on Knowledge and Data Engineering*, vol. 17–6, Jun 2005, pp. 734–749.

[Bal97] Balabanović, M. "An adaptive web page recommendation service". In: Proceedings of the First International Conference on Autonomous Agents, 1997, pp. 378–385.

[Bal98] Balabanović, M. "Exploring Versus Exploiting when Learning User Models for Text Recommendation", *User Modeling and User-Adapted Interaction*, vol. 8–1-2, Mar 1998, pp. 71–102.

[BEL+07] Bennett, J.; Elkan, C.; Liu, B.; Smyth, P.; Tikk, D. "KDD Cup and Workshop", *ACM Special Interest Group on Knowledge Discovery in Data Explorations Newsletter*, vol. 9–2, Dez 2007, pp. 51–52.

[BHK98] Breese, J. S.; Heckerman, D.; Kadie, C. "Empirical analysis of predictive algorithms for collaborative filtering". In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, 1998, pp. 43–52.

[BKV08] Bell, R. M.; Koren, Y.; Volinsky, C. "The bellkor 2008 solution to the netflix prize", *Statistics Research Department at AT&T Research*, vol. 1, 2008.

[BL+07] Bennett, J.; Lanning, S.; et al.. "The netflix prize". In: Proceedings of Knowledge Discovery and Data Mining Cup and Workshop, 2007, pp. 35.

[BOHG13] Bobadilla, J.; Ortega, F.; Hernando, A.; GutiéRrez, A. "Recommender Systems Survey", *Knowledge-Based Systems*, vol. 46, Jul 2013, pp. 109–132.

[BP00] Billsus, D.; Pazzani, M. J. "User modeling for adaptive news access", *User Modeling and User-Adapted Interaction*, vol. 10–2-3, 2000, pp. 147–180.

[Bur02] Burke, R. "Hybrid recommender systems: Survey and experiments", *User Modeling and User-Adapted Interaction*, vol. 12–4, 2002, pp. 331–370.

[CBV10] Cantador, I.; Bellogín, A.; Vallet, D. "Content-based recommendation in social tagging systems". In: Proceedings of the Fourth Conference on Recommender Systems, 2010, pp. 237–240.

[CGM+99] Claypool, M.; Gokhale, A.; Miranda, T.; Murnikov, P.; Netes, D.; Sartin, M. "Combining content-based and collaborative filters in an online newspaper". In: Proceedings of Special Interest Group on Information Retrieval Workshop on Recommender Systems, 1999, pp. 1–11.

[CKT10] Cremonesi, P.; Koren, Y.; Turrin, R. "Performance of Recommender Algorithms on Top-n Recommendation Tasks". In: Proceedings of the Fourth Conference on Recommender Systems, 2010, pp. 39–46.

[CS04] Carenini, G.; Sharma, R. "Exploring more realistic evaluation measures for collaborative filtering". In: Proceedings of the Nineteenth National Conference on Artifical Intelligence, 2004, pp. 749–754.

[CSS99] Cohen, W. W.; Schapire, R. E.; Singer, Y. "Learning to Order Things", *Journal of Artificial Intelligence Research*, vol. 10–1, Maio 1999, pp. 243–270.

[DEC+16] Deldjoo, Y.; Elahi, M.; Cremonesi, P.; Garzotto, F.; Piazzolla, P.; Quadrana, M. "Content-based video recommendation system based on stylistic visual features", *Journal on Data Semantics*, vol. 5–2, 2016, pp. 99–113.

[ERK11] Ekstrand, M. D.; Riedl, J. T.; Konstan, J. A. "Collaborative Filtering Recommender Systems", *Foundation and Trends on Human-Computer Interaction*, vol. 4–2, Fev 2011, pp. 81–173.

[GBC16] Goodfellow, I.; Bengio, Y.; Courville, A. "Deep Learning". MIT Press, 2016, 800p, http://www.deeplearningbook.org.

[GNOT92] Goldberg, D.; Nichols, D.; Oki, B. M.; Terry, D. "Using Collaborative Filtering to Weave an Information Tapestry", *Communications of the ACM*, vol. 35–12, Dez 1992, pp. 61–70.

[Hay08] Haykin, S. O. "Neural Networks and Learning Machines". New York: Pearson, 2008, 3 edition ed., 912p.

[HK16] Harper, F. M.; Konstan, J. A. "The movielens datasets: History and context", *Transactions on Interactive Intelligent Systems*, vol. 5–4, 2016, pp. 19.

[HKTR04] Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; Riedl, J. T. "Evaluating collaborative filtering recommender systems", *Transactions on Information Systems*, vol. 22–1, 2004, pp. 5–53.

[HZRS16] He, K.; Zhang, X.; Ren, S.; Sun, J. "Deep residual learning for image recognition". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[IMG14] IMGUR. "Rory mcilroy swing sequence - 8 frames". Capturado em: http://imgur.com/vhnYIiz, 2017-05-23.

[JLK+15] Jing, Y.; Liu, D.; Kislyuk, D.; Zhai, A.; Xu, J.; Donahue, J.; Tavel, S. "Visual search at pinterest". In: Proceedings of the Twenty-First ACM International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1889–1898.

[JZFF10] Jannach, D.; Zanker, M.; Felfernig, A.; Friedrich, G. "Recommender Systems: An Introduction". New York, NY, USA: Cambridge University Press, 2010, 1st ed., 335p.

[Kar17] Karpathy, A. "Convolutional neural networks". Capturado em: http://cs231n.github.io/convolutional-networks/, 2017-05-10.

[KSH12] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. "Imagenet classification with deep convolutional neural networks". In: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[Lab15] Lab, U. V. "Ilsvrc 2015 results". Capturado em: http://www.image-net.org/challenges/LSVRC/2015/results, 2017-05-10.

[LAK+01] Lawrence, R. D.; Almasi, G. S.; Kotlyar, V.; Viveros, M. S.; Duri, S. S. "Personalization of Supermarket Product Recommendations", *Data Mining and Knowledge Discovery*, vol. 5–1-2, Jan 2001, pp. 11–32.

[LBD+90] LeCun, Y.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. E.; Jackel, L. D. "Handwritten digit recognition with a back-propagation network". In: Advances in Neural Information Processing Systems, 1990, pp. 396–404.

[LDGS11] Lops, P.; De Gemmis, M.; Semeraro, G. "Content-based recommender systems: State of the art and trends". In: *Recommender Systems Handbook*, Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P. B. (Editores), Springer, 2011, cap. 3, pp. 73–105.

[LMY+12] Lü, L.; Medo, M.; Yeung, C. H.; Zhang, Y.-C.; Zhang, Z.-K.; Zhou, T. "Recommender systems", *Physics Reports*, vol. 519–1, 2012, pp. 1–49.

[LR13] LeCun, Y.; Ranzato, M. "A tutorial on deep learning at icml 2013". Capturado em: https://www.slideshare.net/philipzh/a-tutorial-on-deep-learning-at-icml-2013, 2017-05-10.

[MH04] McLaughlin, M. R.; Herlocker, J. L. "A Collaborative Filtering Algorithm and Evaluation Metric That Accurately Model the User Experience". In: Proceedings of the Twenty-Seventh Annual International Conference on Research and Development in Information Retrieval, 2004, pp. 329–336.

[MKP03] Mak, H.; Koprinska, I.; Poon, J. "INTIMATE: a Web-based movie recommender using text categorization". In: Proceedings of the IEEE International Conference on Web Intelligence, 2003, pp. 602–605.

[NDRV09] Nanas, N.; De Roeck, A.; Vavalis, M. "What happened to content-based information filtering?" In: Conference on the Theory of Information Retrieval, 2009, pp. 249–256.

[RDS+15] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al.. "Imagenet large scale visual recognition challenge", *International Journal of Computer Vision*, vol. 115–3, 2015, pp. 211–252.

[RRS11] Ricci, F.; Rokach, L.; Shapira, B. "Introduction to recommender systems handbook". In: *Recommender Systems Handbook*, Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P. B. (Editores), Springer-Verlag New York, Inc., 2011, cap. 1, pp. 1–35.

[RSS05] Rasheed, Z.; Sheikh, Y.; Shah, M. "On the use of computable features for film classification", *Transactions on Circuits and Systems for Video Technology*, vol. 15–1, 2005, pp. 52–64.

[RV97] Resnick, P.; Varian, H. R. "Recommender Systems", *Communications of the ACM*, vol. 40–3, Mar 1997, pp. 56–58.

[SC00] Smyth, B.; Cotter, P. "A personalised TV listings service for the digital TV age", *Knowledge-Based Systems*, vol. 13–2, 2000, pp. 53–59.

[SG11] Shani, G.; Gunawardana, A. "Evaluating recommendation systems". In: *Recommender Systems Handbook*, Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P. B. (Editores), Springer, 2011, cap. 8, pp. 257–297.

[SKB+98] Sarwar, B. M.; Konstan, J. A.; Borchers, A.; Herlocker, J.; Miller, B.; Riedl, J. "Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system". In: Proceedings of the ACM Conference on Computer Supported Cooperative Work, 1998, pp. 345–354.

[SKKR00] Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. "Analysis of recommendation algorithms for e-commerce". In: Proceedings of the Second Conference on Electronic Commerce, 2000, pp. 158–167.

[SKKR01] Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J. "Item-based Collaborative Filtering Recommendation Algorithms". In: Proceedings of the Tenth International Conference on World Wide Web, 2001, pp. 285–295.

[SLJ+15] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. "Going deeper with convolutions". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[SRASC14] Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. "Cnn features off-the-shelf: An astounding baseline for recognition". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 806–813.

[SWBR16] Simões, G. S.; Wehrmann, J.; Barros, R. C.; Ruiz, D. D. "Movie genre classification with convolutional neural networks". In: International Joint Conference on Neural Networks, 2016, pp. 259–266.

[TSK05] Tan, P.-N.; Steinbach, M.; Kumar, V. "Introduction to Data Mining, (First Edition)". Addison-Wesley Longman Publishing Co., Inc., 2005, 769p.

[VdODS13] Van den Oord, A.; Dieleman, S.; Schrauwen, B. "Deep content-based music recommendation". In: Advances in Neural Information Processing Systems, 2013, pp. 2643–2651.

[YBL16] You, Q.; Bhatia, S.; Luo, J. "A picture tells a thousand words—About you! User interest profiling from user generated visual content", *Signal Processing*, vol. 124, 2016, pp. 45–53.

[ZLX+14] Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; Oliva, A. "Learning deep features for scene recognition using places database". In: Advances in Neural Information Processing Systems, 2014, pp. 487–495.

[ZMKL05] Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; Lausen, G. "Improving recommendation lists through topic diversification". In: Proceedings of the Fourteenth International Conference on World Wide Web, 2005, pp. 22–32.