



TRANSFORMANDO O PROBLEMA DE FILTRAGEM COLABORATIVA EM
APRENDIZADO DE MÁQUINA SUPERVISIONADO

Filipe Braida do Carmo

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Zimbrão da Silva

Rio de Janeiro
Fevereiro de 2013

TRANSFORMANDO O PROBLEMA DE FILTRAGEM COLABORATIVA EM
APRENDIZADO DE MÁQUINA SUPERVISIONADO

Filipe Braida do Carmo

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Geraldo Zimbrão da Silva, D.Sc.

Prof. Jano Moreira de Souza, Ph.D.

Prof. Alexandre Plastino de Carvalho, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

FEVEREIRO DE 2013

Carmo, Filipe Braida do

Transformando o problema de filtragem colaborativa em aprendizado de máquina supervisionado/ Filipe Braida do Carmo. – Rio de Janeiro: UFRJ/COPPE, 2013.

XIV, 94 p.: il.; 29,7 cm.

Orientador: Geraldo Zimbrão da Silva

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2013.

Referencias Bibliográficas: p. 88-94.

1. Sistemas de Recomendação. 2. Redução de Dimensionalidade. 3. Aprendizado de Máquina. I. Silva, Geraldo Zimbrão. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III Título.

À minha família.

AGRADECIMENTOS

Estes três anos em que fiz esta pesquisa foram uma árdua jornada de desafio, construção e amadurecimento.

Meus respeitosos agradecimentos pela contribuição da banca e especialmente aos meus orientadores professores Geraldo Zimbrão e Carlos Eduardo Mello. Não me esquecerei dos seus ensinamentos, conselhos e sua inestimável confiança. Não somente no âmbito deste trabalho, mas também durante nos projetos e na vida.

Agradeço especialmente a minha noiva Mariana por estar sempre ao meu lado me apoiando e ensinando-me a nunca desistir dos meus sonhos através de todo seu amor e carinho. Sou eternamente grato.

Agradeço aos meus companheiros de laboratório Marden, Luiz Orleans, Fellipe Duarte e Pedro. Eles acompanharam desde o início e ajudaram muito no desenvolvimento deste trabalho.

Também agradeço a todos os meus amigos. Principalmente ao Roque, Cláudio e o David. Sem a amizade e o apoio deles não estaria onde eu estou e posso dizer que cresci muito como pessoa graças à amizade deles.

Por fim, agradeço em especial àqueles que sempre me apoiaram, ensinando os valores da educação e da ética ao longo da minha vida: minha família.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

TRANSFORMANDO O PROBLEMA DE FILTRAGEM COLABORATIVA EM
APRENDIZADO DE MÁQUINA SUPERVISIONADO

Filipe Braida do Carmo

Fevereiro/2013

Orientador: Geraldo Zimbrão da Silva

Programa: Engenharia de Sistemas e Computação

Filtragem colaborativa é uma abordagem conhecida em sistemas de recomendação, que tem como o objetivo de prever as notas de usuários sobre itens utilizando avaliações de outros. Devido à dificuldade de representar em um mesmo espaço de características usuários e itens, torna-se difícil construir um conjunto de treinamento a partir do qual possa se obter um modelo de aprendizado supervisionado. Diversos métodos vêm sendo propostos na literatura com objetivo de atacar esse problema, *i.e.*, construir um espaço comum de características com base nas informações do domínio, de modo a permitir a utilização dessas técnicas. Entretanto, não existe um método que faça essa tarefa independentemente de domínio. Nesse contexto, o objetivo deste trabalho consiste em propor uma metodologia para a transformação da tarefa de filtragem colaborativa em aprendizado supervisionado de máquina. Esta proposta baseia-se na construção de um espaço de características de forma a representar as avaliações de usuários sobre itens a partir da extração das variáveis latentes dos dados originais. Esta proposta foi avaliada sob um conjunto de dados reais, apresentando resultados superiores aos métodos clássicos da literatura.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

TRANSFORMING COLLABORATIVE FILTERING INTO SUPERVISED MACHINE
LEARNING PROBLEM

Filipe Braida do Carmo

February/2013

Advisor: Geraldo Zimbrão da Silva

Department: Computer Science Engineering

Collaborative filtering, a well-known approach for recommender system, aims at predicting ratings of users for items based on items previously rated by others. Due to the difficulty to represent ratings in a feature space comprehending both users and items, it becomes a tough task to build training sets from which supervised learning models might be obtained. There have been several methods in the literature tackling this problem, *i.e.*, building a common space of feature for collaborative filtering. However, these proposals usually rely on domain information, which might be not available in all settings. In this context, this work aims at proposing a methodology to transform the classic collaborative filtering setting into the supervised machine learning problem. This proposal relies on building a feature space from the latent variables hidden on the original data. Experiments have been performed in order to evaluate the proposal. The obtained results have shown satisfactory performance of the proposed algorithms over classic collaborative filtering methods.

ÍNDICE

Capítulo 1 – Introdução	1
1.1 Motivação	1
1.2 Objetivos	3
1.3 Hipótese	3
1.4 Organização	3
Capítulo 2 – Fundamentação Teórica	5
2.1 - Sistemas de Recomendação	5
2.1.1 - Abordagens de Recomendação	8
2.2 – Aprendizado de Máquina	22
2.2.1 – Aprendizado Supervisionado	25
2.3 – Trabalhos Relacionados	38
Capítulo 3 – COFISL (<i>Collaborative Filtering to Supervised Learning</i>)	41
3.1 – Introdução	41
3.2 – Definição do Problema	43
3.3 – Proposta	44
3.3.1 – Transformação	46
Capítulo 4 – Avaliação Experimental	54
4.1 - Objetivos dos Experimentos	54
4.2 - Base de Dados	54
4.3 - Metodologia e Organização dos Experimentos	57
4.3 - Avaliação	62
4.4 – Resultados	63
4.4.1 – Experimento I	63
4.4.2 – Experimento II	66
4.4.3 – Experimento III	74
4.4.4 – Experimento IV	77
4.5 - Análise dos Dados	79
Capítulo 5 – Conclusões	85
5.1 Considerações acerca do trabalho	85
5.2 Contribuições	85
5.3 Limitações e trabalhos futuros	86

LISTAGEM DE FIGURAS

Figura 1 - Exemplo de itens que foram avaliados em comum para o cálculo da similaridade. (SARWAR <i>et al.</i> , 2001)	13
Figura 2- Ilustração simplificada da ideia das variáveis latentes onde é utilizado quatro fatores, ou melhor, categorias para caracterizar o usuário e o item (nesse caso um filme). (KOREN <i>et al.</i> , 2009)	16
Figura 3 – Imagem acima a decomposição SVD. A imagem abaixo a decomposição aproximada.	16
Figura 4 – Um conjunto de dados e os pontos plotados no gráfico. (MARSLAND, 2009)	23
Figura 5 - Processo de Aprendizagem – (Própria Autoria)	24
Figura 6 – Processo de Aprendizado Supervisionado.	26
Figura 7 - Dois gráficos exemplificando a fronteira de decisão. No lado esquerdo utilizando retas e no lado direito utilizando curvas.	28
Figura 8 - Exemplo de uma Árvore de Decisão.	31
Figura 9 - Florestas Aleatórias. (TAN <i>et al.</i> , 2005)	33
Figura 10 - Exemplo do K-NN com k valendo 1, 2 e 3 respectivamente.	34
Figura 11 - - Exemplo do K-NN sendo que o k grande.	35
Figura 12 - Componentes do neurônio biológico.	36
Figura 13 - Modelando uma função booleana usando um perceptron. (TAN <i>et al.</i> , 2005)	37
Figura 14 - Arquitetura de uma rede de perceptron utilizando duas camadas.	37
Figura 15 - Exemplo de conversão da matriz de preferências para a matriz de gosto utilizado por (BILLSUS, 1998).	39
Figura 16 – Aprendizado de máquina supervisionado.	42
Figura 17 – Processo de transformação da matriz de notas para um classificador ou regressor.	44
Figura 18 – Proposta	46

Figura 19 - Representação do modelo de recomendação através de um grafo bipartido não direcionado.	47
Figura 20 - Exemplo de substituição dos valores desconhecidos por zero.	48
Figura 21 – Decomposição utilizando a técnica SVD extraindo dois fatores.	50
Figura 22 – Representação gráfica do espaço gerado pelos fatores do usuário (à esquerda) e do item (à direita).	50
Figura 23 – Exemplo da proposta de transformação.	52
Figura 24 – Representação visual do espaço do conjunto de treinamento gerado no exemplo anterior.	52
Figura 25 - Representação visual do espaço do conjunto de treinamento gerado no exemplo anterior utilizando a média do usuário.	53
Figura 26 - Quantidade de Avaliações com relação a nota da base <i>MovieLens</i> .	55
Figura 27 - Quantidade de Avaliações com relação ao usuário da base <i>MovieLens</i> .	56
Figura 28 - Quantidade de Avaliações com relação ao filme da base <i>MovieLens</i> .	56
Figura 29 – Exemplo de pré-processamento utilizar a normalização pelo usuário.	57
Figura 30 – Quantidade de informação que cada variável latente possui.	60
Figura 31 – Ganho de informação que cada variável latente possui.	60
Figura 32 – Avaliação do MAE da técnica ANN variando a quantidade de neurônios.	64
Figura 33 - Avaliação do RMSE da técnica ANN variando a quantidade de neurônios.	64
Figura 34 - Avaliação do MAE da técnica floresta aleatória variando a quantidade de árvores de decisões.	65
Figura 35 - Avaliação do RMSE da técnica floresta aleatória variando a quantidade de árvores de decisões.	65
Figura 36 – MAE do experimento utilizando Naive Bayes e sem normalização.	67
Figura 37 – RMSE do experimento utilizando Naive Bayes e sem normalização.	67
Figura 38 - MAE do experimento utilizando Rede Neural e sem normalização.	68

Figura 39 - RMSE do experimento utilizando Rede Neural e sem normalização.	68
Figura 40 - MAE do experimento utilizando Rede Neural e com normalização através da média do item.	69
Figura 41 - RMSE do experimento utilizando Rede Neural e com normalização através da média do item.	69
Figura 42 - MAE do experimento utilizando Rede Neural e com normalização através da média do usuário.	70
Figura 43 - RMSE do experimento utilizando Rede Neural e com normalização através da média do usuário.	70
Figura 44 - MAE do experimento utilizando Floresta Aleatória e sem normalização.	71
Figura 45 - RMSE do experimento utilizando Floresta Aleatória e sem normalização.	71
Figura 46 - MAE do experimento utilizando Floresta Aleatória e com normalização através da média do item.	72
Figura 47 - RMSE do experimento utilizando Floresta Aleatória e com normalização através da média do item.	72
Figura 48 - MAE do experimento utilizando Floresta Aleatória e com normalização através da média do usuário.	73
Figura 49 - RMSE do experimento utilizando Floresta Aleatória e com normalização através da média do usuário.	73
Figura 50 - MAE dos algoritmos baseados em memória e comparando com o melhor configuração da proposta.	75
Figura 51 - RMSE dos algoritmos baseados em memória e comparando com o melhor configuração da proposta.	75
Figura 52 – Cobertura dos algoritmos baseados em memória.	76
Figura 53 - MAE dos algoritmos baseados em modelo e comparando com o melhor configuração da proposta.	77

Figura 54 - RMSE dos algoritmos baseados em modelo e comparando com o melhor configuração da proposta.	77
Figura 55 - MAE do experimento utilizando a melhor configuração do experimento II variando o número de vizinhos.	78
Figura 56 - RMSE do experimento utilizando a melhor configuração do experimento II variando o número de vizinhos.	79

LISTAGEM DE TABELAS

Tabela 1 - Fragmento da matriz de notas de um sistema de recomendação de filmes.	7
Tabela 2 - Fragmento da matriz de notas de um sistema de recomendação de filmes.	11
Tabela 3 – Valor de uma residência por seu respectivo tamanho.	28
Tabela 4 - Conversão da filtragem colaborativa para aprendizado supervisionado.	44
Tabela 5 – Os melhores resultados (MAE e RMSE) do experimento dois e seus respectivas configurações.	80
Tabela 6 - Comparação da proposta (COFISL) com as técnicas clássicas da área de sistema de recomendação.	81
Tabela 7 – Filtragem Colaborativa baseado no usuário utilizando a similaridade cosseno e o mínimo de treze vizinhos.	83
Tabela 8 – Matriz de confusão da proposta utilizando o algoritmo floresta aleatória, SVD, normalização através da média do usuário e oito variáveis latentes.	83

Capítulo 1 – Introdução

1.1 Motivação

Com o advento da internet no último século, a sociedade moderna vem produzindo informação como nunca houve. Em toda a história da humanidade foram produzidos pelo menos 32 milhões de livros, 750 milhões de artigos, 25 milhões de músicas, 500 milhões de imagens, 500 mil filmes, três milhões de vídeos e programas de TV, e cerca de 100 bilhões de páginas na web, sendo que, a maior parte dessa produção é referente aos últimos 50 anos de nossa história. Consideremos ainda que o volume de informação gerado por dia dobra a cada cinco anos, o que fará com que esses números cresçam ainda mais nas próximas décadas. (TAPSCOTT & WILLIAMS, 2006)

Essa grande quantidade de informação e conhecimento gerados nessas últimas décadas vem mudando a estrutura da sociedade e tornando-se o fator de poder e mudança social. O indivíduo, dentro dessa sociedade de informação, precisa ser capaz de absorver e filtrar essa grande quantidade de informação para que assim consiga construir novo conhecimento. Contudo, filtrar essas informações é uma tarefa difícil, porque normalmente estas não estão estruturadas (BAEZA-YATES & RIBEIRO-NETO, 1999) e para avaliar qual o seu grau de importância seria necessário identificar todas as informações para tal.

Dentro da *Word Wide Web* (WWW), o problema da grande quantidade de informação se agrava. Em outubro de 2009, o número de páginas era em torno de 35 bilhões (WORLDWEBSIZE, 2012). Esse valor inviabiliza a avaliação individual de cada página ou informação na web. Por causa desse problema, no meio da década de 90 surge a área de sistemas de recomendação. Esta tem como motivação ajudar usuários com a lidar com a enorme quantidade de dados através de recomendações personalizadas de conteúdo e/ou de serviços. Diversas empresas têm adotado esse tipo de tecnologia como exemplo, a *Amazon*¹ com livros, *Netflix*² com filmes e a *Google*³ com notícias.

¹ <http://www.amazon.com>

² <http://www.netflix.com>

³ <http://www.google.com>

A recomendação pode ser vista no dia a dia, como por exemplo, a ida a uma loja de roupa. Dentro dela existe uma enorme quantidade de roupas, inviabilizando a avaliação de todas as peças disponíveis. Para encontrar a informação adequada, *i.e.*, uma peça de roupa apropriada, a atendente da loja realiza um processo natural de filtragem e de recomendação. Desta forma, o comprador torna-se mais suscetível a comprar, pois o número de opções de escolha foi reduzido de maneira drástica.

Um sistema de recomendação é definido como um sistema que consegue avaliar os gostos dos usuários por um conjunto de itens e assim predizer quais os itens o usuário não possui o conhecimento ou não tenha consumido gostará. A forma de realizar a recomendação pode ser feita de diversas formas. Na literatura existem diversas abordagens para a realização da recomendação podendo, por exemplo, utilizar as informações dos itens para realizar a tarefa (baseada em conteúdo) (ADOMAVICIUS & TUZHILIN, 2005) ou utilizar as preferências explícitas dos usuários sobre os itens (filtragem colaborativa) (GOLDBERG *et al.*, 1992).

Contudo existem diversas limitações na aplicação dos algoritmos baseados em filtragem colaborativa dentro do domínio de sistemas de recomendações. Os algoritmos são baseados nos itens avaliados que são comuns entre usuários, ou usuários comuns entre itens, e normalmente essas notas são poucas em comparado a quantidade de usuários e itens dentro do sistema (SARWAR *et al.*, 2000). Outro problema é que essa matriz é dinâmica, ou seja, não existe uma quantidade fixa de usuários ou itens e quando existisse um novo usuário ou item não seria capaz de realizar a recomendação (PAPAGELIS *et al.*, 2005)

O sistema de recomendação é um problema da área de reconhecimento de padrões (*pattern recognition*), ou seja, através de observações de padrões ou de conhecimentos a priori tentar inferir informação. O valor inferido pode ser um valor discreto (classificação) ou contínuo (regressão). Esse problema é discutido dentro da área de aprendizado de máquina podendo ser caracterizado dependendo da forma do aprendizado como aprendizado supervisionado ou não supervisionado. (BISHOP, 2006)

Entretanto, por causa das características do domínio de sistema de recomendação discutidas anteriormente, não é possível aplicar diretamente as técnicas de aprendizado supervisionado. Existe na literatura alterações nos algoritmos para que ele consiga ser capaz

de lidar com a matriz de usuário por item ou utilizar as informações sobre o domínio para poder transformar a matriz em um conjunto de treinamento típico de um treinamento de uma máquina de aprendizado supervisionado.

Esse trabalho tem como objetivo propor uma metodologia para a aplicação das técnicas de aprendizado de máquina de forma genérica, ou seja, somente utilizando as informações sobre as avaliações do usuário sobre o item sem utilizar as informações do domínio da aplicação, e estudar o problema de sistema de recomendação como um problema típico de aprendizado de máquina.

1.2 Objetivos

Os objetivos deste trabalho são:

- Propor uma metodologia para a transformação do problema de filtragem colaborativa em aprendizado de máquina supervisionado;
- Implementação dessa transformação utilizando diversas técnicas de aprendizado supervisionado;
- Comparação dos desempenhos encontrados utilizando a metodologia proposta com as técnicas clássicas da literatura de sistemas de recomendação.

1.3 Hipótese

Os sistemas de recomendação surgiram para ajudar o usuário com a tarefa de lidar com as grandes quantidades de informação e providenciar recomendações personalizadas de conteúdo ou de serviços.

Assim, a hipótese deste trabalho é:

- É possível transformar o problema de filtragem colaborativa em um problema clássico de aprendizado supervisionado.

1.4 Organização

Esta dissertação está organizada da seguinte maneira:

O capítulo 2 discute conceitos referentes à fundamentação teórica do trabalho. São abordados os conceitos de sistemas de recomendação e aprendizado de máquina;

No capítulo 3 será apresentada a proposta para transformar o problema de sistema de recomendação em um problema de aprendizado de máquina;

O capítulo 4 formalizará a metodologia para realização dos experimentos e o seu resultado;

Por fim, são apresentadas a conclusão e as referências bibliográficas deste trabalho.

Capítulo 2 – Fundamentação Teórica

Para que seja possível o entendimento das técnicas e tecnologias utilizadas, este capítulo apresenta os conceitos teóricos de maior relevância. As seções tratam os seguintes assuntos: sistemas de recomendação, onde será apresentado esse domínio de aplicações e as técnicas mais clássicas da área; Aprendizado de Máquina, nesta seção será detalhado um conjunto de técnicas e algoritmos que serão utilizados neste presente trabalho; Por fim, serão mencionados os trabalhos na literatura que utilizam esse conjunto de técnicas dentro do contexto da recomendação.

2.1 - Sistemas de Recomendação

Durante toda a história, as pessoas sempre recorreram às recomendações com o objetivo de facilitar ou minimizar o risco de uma tomada de decisão. A importância da recomendação cresceu com o surgimento da sociedade de informação, na qual a informação ganhou uma grande importância e se tornou o fator de poder e de mudança social. Essa importância se deve ao desenvolvimento e o barateamento das tecnologias de informação e de comunicação (TIC).

A Internet é o apogeu do desenvolvimento das TIC's e seu impacto foi tão grande que passou a ter um papel central nessa sociedade de informação, tanto no armazenamento das informações quanto em termos de circulação de capital e das relações sociais. Um dos grandes problemas da internet é o seu tamanho, o que dificulta a busca da informação.

Assim, surgiu no início dos anos 90 o primeiro Sistema de Recomendação (SR), o *Tapestry* (GOLDBERG *et al.*, 1992). Esse trabalho vislumbrou a criação do termo filtragem colaborativa (*collaborative filtering*) que designa um sistema que utiliza os relacionamentos entre os usuários pela forma que o mesmos avaliaram um conjunto de itens. Esse termo se tornou referência para uma família de técnicas que utilizam a extração das informações dos usuários pela forma que avaliam os itens ou vice-versa.

O *Tapestry* foi desenvolvido pela *Xerox Palo Alto* e tem como objetivo a seleção de e-mail. Os usuários criam, através de uma interface, regras (filtros) que relacionam suas preferências. Através desses filtros os e-mails são selecionados e somente serão enviados para o usuário aqueles que estiverem de acordo ao respectivo usuário.

Devido à simplicidade dos algoritmos utilizando a filtragem colaborativa, ela acabou sendo amplamente utilizada para fins comerciais e atualmente possui uma grande importância nesse contexto. Estudos têm demonstrado que SR trazem três principais benefícios para o comércio eletrônico: o aumento das vendas, vendas cruzadas e uma maior lealdade dos seus usuários. Diversos sites como *Amazon*⁴, *Netflix*⁵ e *Youtube*⁶, utilizam a recomendação como diferencial no negócio. (LINDEN *et al.*, 2003)

(ADOMAVICIUS & TUZHILIN, 2005) descreveu com uma maior formalidade o problema de recomendação como: Dado U o conjunto de todos os usuários e I o conjunto de todos os itens como exemplo livros, músicas ou vídeos. Dado r a função utilidade que informa a importância do item i ao usuário u , i.e, $r: U \times I \rightarrow R$, onde $R \in \mathbb{N}$ ou $R \in \mathbb{R}$, é o conjunto de preferência do usuário pelo item. Então, para cada usuário $u \in U$ é escolhido o item i' onde maximize a função utilidade r , ou seja:

$$\forall u \in U, i' = \operatorname{argmax} r(u, i), \quad \text{sendo } i' \in I$$

No contexto de sistema de recomendação, um usuário seleciona um valor dentro do conjunto de preferência (R). Esse é um valor significativo para um usuário para demonstrar o quanto ele gostou de um item particular. Na maioria dos SR o conjunto de preferência é dado por uma nota, como exemplo, um usuário indicou uma nota 6 (no máximo até 10) ao filme “Titanic”. Dependendo do sistema esse conjunto pode variar inclusive podendo ser um valor real até uma classe binária (“gostou” e “não gostou”). Na Tabela 1 é demonstrado um exemplo de uma matriz de preferências de usuário-item utilizando o conjunto de preferência gostou e não gostou.

⁴ <http://www.amazon.com>

⁵ <http://www.netflix.com>

⁶ <http://www.youtube.com>

Tabela 1 - Fragmento da matriz de notas de um sistema de recomendação de filmes.

	Titanic	Poderoso Chefão	Matrix	Toy Story
Cláudio	Não Gostou	∅	Gostou	Gostou
Roque	∅	Gostou	∅	Não Gostou
David	Gostou	∅	∅	∅

O problema central de um sistema de recomendação é que normalmente a função utilitária u não é definida em todo espaço $U \times I$, mas em um subconjunto deste. Como exemplo a Tabela 1 possui o conjunto R com dois valores {"gostou", "não gostou"} e o símbolo "∅" que significa que o usuário não explicitou sua preferência sobre esse item. O objetivo do recomendador é estimar os valores não avaliados pelo usuário e através dessas estimativas, oferecer boas recomendações.

Existem duas abordagens possíveis para estimar a preferência de um usuário por um item que não está explicitado. A primeira é utilizando heurísticas para definir a função utilitária u . O outro método é a utilização de uma função que minimiza algum critério de desempenho, como exemplo a raiz quadrada da média do erro (RMSE). (ADOMAVICIUS & TUZHILIN, 2005)

Após estimar todos os valores não avaliados pelos usuários utilizando alguma das duas abordagens citadas acima, o sistema de recomendação selecionará o item cujo valor estimado possua o maior significado dentro do conjunto de preferência, como exemplo a maior nota, para ser recomendado para o usuário. Podendo também ser recomendado o conjunto de N melhores itens para o usuário.

A estimativa da nota de um item que não foi avaliado por um usuário pode ser feita de diversas formas como utilizando técnicas de aprendizado de máquina, teoria da aproximação, e diversas outras heurísticas. A abordagem que é realizada para estimar uma nota é utilizada para classificar os algoritmos.

Em (BURKE, 2007), foi proposta uma taxonomia de cinco diferentes classes e o trabalho foi estendido por (RICCI *et al.*, 2011) adicionando uma nova classe:

- Baseada em conteúdo: A recomendação se dá através de itens similares aos do conjunto de preferência do usuário;
- Filtragem Colaborativa: A recomendação é através dos itens de outros usuários similares, ou seja, com gostos e preferências parecidas ao usuário;
- Demográfico: O sistema utiliza a informação demográfica do usuário para realizar recomendação;
- Baseada no conhecimento: A recomendação é baseada inferindo sobre as necessidades dos usuários e suas preferências utilizando o conhecimento sobre o domínio.
- Baseada na comunidade: A recomendação é feita utilizando as preferências dos usuários relacionados;
- Abordagens híbridas: Método que combina algumas das abordagens citadas acima.

2.1.1 - Abordagens de Recomendação

Nesta seção será apresentado um maior detalhamento das três principais abordagens de sistemas de recomendação (filtragem colaborativa, baseado em conteúdo e híbrida) e os seus principais algoritmos.

2.1.1.1 - Filtragem Colaborativa (*Collaborative Filtering*)

No dia a dia, as pessoas recorrem aos outros em busca de recomendações para obterem auxílio no processo de tomada decisão. A filtragem colaborativa utiliza como base esse conceito de utilizar a opinião das outras pessoas para recomendação, ou seja, utilizando o conjunto de preferência dos usuários para essa tarefa. Para tal, podemos identificar um subconjunto de usuários similares a u que explicitaram alguma nota para o item i . Através deste subconjunto podemos prever a preferência de um usuário u sobre o item i .

A ideia central dessa abordagem é identificar o conjunto de usuários V que possuem o mesmo perfil de comportamento do que o usuário u e explicitaram a nota para o item i . Supondo que o usuário u tenda para o mesmo comportamento do conjunto V , podemos afirmar que a nota dele para o item i será dada pelas as notas dadas por V . Devido às suas características, a filtragem colaborativa também é chamada na literatura de abordagem social.

A filtragem colaborativa possui diversos benefícios sobre a baseada em conteúdo. Ela não exige nenhum conhecimento adicional além da própria dinâmica da recomendação. Na outra abordagem existe a necessidade de ter a informação sobre o item e a recomendação está diretamente relacionada com a qualidade desta. Outra desvantagem é que a filtragem colaborativa tem a liberdade de recomendar itens com conteúdos diferentes, com isso evitando o problema de recomendar “mais dos mesmos”, ou seja, aumentando as recomendações *serendipity* (GE *et al.*, 2010).

Entretanto existem diversos desafios nesta abordagem. Em geral, os sistemas de recomendações possuem uma base grande de produtos. Isso torna a matriz de usuário-item extremamente esparsa e a qualidade e o desempenho da recomendação está diretamente relacionado com o tamanho e a esparsidade dos dados (LINDEN *et al.*, 2003). Normalmente a quantidade de dados em um sistema de recomendação comparados à matriz de usuário-item completa é cerca de 1% (SARWAR *et al.*, 2001).

Outro problema decorrente da esparsidade dos dados é o surgimento de um usuário ou item novo. Devido às características intrínsecas da filtragem colaborativa, a entrada de um novo usuário ou item torna a tarefa difícil de encontrar o conjunto de usuários similares. Com isso impossibilita a recomendação desse item até o momento que uma quantidade de usuários o recomende ou recomendar itens para um usuário até que ele recomende uma quantidade de itens. Esse problema é chamado na literatura de *cold-starter* (SCHEIN *et al.*, 2002).

Na literatura existem diversas abordagens que tentam minimizar os efeitos dos problemas descritos anteriormente. A maioria dessas faz o uso de técnicas de redução de dimensionalidade como a decomposição de valores singulares (SVD) ou análise das principais componentes (PCA) para lidar com a esparsidade e com isso prover boas recomendações. (BILLSUS, 1998, GOLDBERG *et al.*, 2001, RICCI *et al.*, 2011)

Mesmo com esses desafios, a filtragem colaborativa obteve um grande sucesso tanto na pesquisa quanto na prática (SCHAFER *et al.*, 1999, HUANG & GONG, 2008, SARWAR *et al.*, 2002). No entanto, ainda existem diversas questões a serem pesquisadas para superar os desafios intrínsecos à filtragem colaborativa como a esparsidade e a escalabilidade. Para tal, a literatura divide essa abordagem em duas grandes classes: baseada em memória e modelo (ADOMAVICIUS & TUZHILIN, 2005).

2.1.2.2.1 - Memória

Os algoritmos baseados em memória, ou vizinhos mais próximos, pressupõem que os usuários podem ser agrupados pela similaridade. Identificando os vizinhos mais próximos de um usuário u , o algoritmo utiliza diretamente a preferência destes sobre o item i para realizar a predição. Devido à utilização da similaridade como heurística essa classe de algoritmo também é chamado na literatura como algoritmos baseados em heurísticas (ADOMAVICIUS & TUZHILIN, 2005).

Dentro dessa classe de algoritmos existem duas formas de realizar a predição dependendo do agrupamento realizado. O primeiro é baseado no usuário, onde o sistema utiliza os usuários similares ao usuário u que avaliaram o item i e com isso predizer a preferência do usuário u para o item i . Por último o baseado no item tenta predizer a nota de um item i para o usuário u baseado nas preferências do u para os itens similares a i .

Essa abordagem pode ser interpretada como uma automatização da recomendação boca-a-boca. As pessoas possuem o hábito de obter recomendações e opiniões sobre produtos ou serviços (livros, filmes, artigos, restaurantes, etc.) de outros indivíduos que tenham consumido estes e que seja considerada uma fonte confiável. Através dessas opiniões a pessoa poderia ponderar com relação à confiança dela pela fonte da recomendação, e sua expectativa sobre o produto ou serviço.

Para ilustrar, considere o exemplo da Tabela 2. Ela é um fragmento da matriz de notas de um sistema de recomendação de filmes com o conjunto de preferência definido por 1 a 5. A usuária Mariana deseja saber se aluga o filme Matrix e verificou que o Cláudio e o Roque gostaram do filme e o usuário David não gostou. Como ela sabe que o Cláudio e o Roque, diferentemente do David, possuem os gostos parecidos aos dela podemos dizer que ela possui grandes chances de gostar do filme.

Tabela 2 - Fragmento da matriz de notas de um sistema de recomendação de filmes.

	Titanic	Poderoso Chefão	Matrix	Toy Story
Cláudio	4	3	5	4
Roque	3	4	4	4
David	2	4	1	1
Mariana	4	5	?	4

Recomendações baseadas nos usuários são métodos que tentam prever a preferência de um usuário u para um item i , ou seja, r'_{ui} . Definindo a similaridade de dois usuários u e v como w_{uv} e supondo que o item i já foi avaliado por um conjunto de usuários. Definindo $\aleph_i(u)$ como k vizinhos mais próximos do usuário u que avaliaram o item i . Assim, a previsão será dada pela média ponderada das notas dos usuários similares a u como mostra a Equação 1.

$$r'_{ui} = \frac{\sum_{v \in \aleph_i(u)} w_{uv} r_{vi}}{\sum_{v \in \aleph_i(u)} |w_{uv}|}$$

Equação 1 – Recomendação pelos vizinhos mais próximos baseado no usuário.

Recomendações baseadas nos itens são métodos que tentam prever a preferência de um item i para usuário u , ou seja, r'_{ui} . Definindo a similaridade de dois itens i e j como w_{ij} e supondo que o usuário u já avaliou diversos itens. Definindo $\aleph_u(i)$ como k vizinhos mais próximos do item i que foram avaliados pelo usuário u . Assim, a previsão será dada pela média ponderada das notas dos itens similares a i como mostra a Equação 2.

$$r'_{ui} = \frac{\sum_{j \in \aleph_u(i)} w_{ij} r_{uj}}{\sum_{j \in \aleph_u(i)} |w_{ij}|}$$

Equação 2 - Recomendação pelos vizinhos mais próximos baseado no item.

Um problema que ocorre na prática é que os usuários possuem formas diferentes de avaliar os itens, ou seja, o significado do conjunto de preferência possui significados diferentes para cada usuário. Tanto na Equação 1 e na Equação 2 esse fator não é considerado

e com isso prejudica a previsão. Uma solução para esse problema é utilizar as notas normalizadas ($h(r_{ij})$) para realizar a previsão (RICCI *et al.*, 2011), como mostra a Equação 3:

$$r'_{ui} = h^{-1} \left(\frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} h(r_{vi})}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \right) \quad r'_{ui} = h^{-1} \left(\frac{\sum_{j \in \mathcal{N}_i(u)} w_{ij} h(r_{uj})}{\sum_{j \in \mathcal{N}_i(u)} |w_{ij}|} \right)$$

Equação 3 – As equações normalizadas. À esquerda baseada no usuário e à direita baseada no item.

Na literatura, a normalização mais utilizada é a média central. O objetivo da média central é utilizar a variação em relação à média das notas do usuário ou do item como o valor na qual se deseja prever. Um fator interessante da média central é que o fator de apreciação do usuário para um item é se a nota normalizada ($h(r_{ui})$) é positiva ou negativa. (RICCI *et al.*, 2011)

$$h(r_{ui}) = r_{ui} - \bar{r}_u \quad h(r_{ui}) = r_{ui} - \bar{r}_i$$

Equação 4 – Normalização pela média central. À esquerda utilizando a média do usuário e à direita a do item.

Simplificando as equações ficamos com:

$$r'_{ui} = \bar{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} (r_{vi} - \bar{r}_v)}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|} \quad r'_{ui} = \bar{r}_i + \frac{\sum_{j \in \mathcal{N}_i(u)} w_{ij} (r_{uj} - \bar{r}_j)}{\sum_{j \in \mathcal{N}_i(u)} |w_{ij}|}$$

Equação 5 – Recomendação normalizada utilizando média central. À esquerda baseado no usuário e à direita baseado no item.

O fator chave na recomendação por vizinhos mais próximos é a similaridade. Ela é a responsável pela seleção dos vizinhos mais confiáveis que serão utilizados para a previsão e além de fornecer o grau de importância desses vizinhos. Devido a esses fatores, a escolha do método de similaridade impacta na precisão e no desempenho do algoritmo de recomendação. (RICCI *et al.*, 2011)

Existem várias abordagens para o cálculo da similaridade entre dois usuários ou dois itens. Na maioria dos casos, a similaridade é baseada nas notas que foram avaliadas em comum, ou seja, co-avaliadas entre dois usuários ou dois itens (ADOMAVICIUS & TUZHILIN, 2005). Existem dois métodos comuns na literatura: a correlação baseada em Pearson (SHARDANAND & MAES, 1995) e pelo Cosseno (LINDEN *et al.*, 2003).

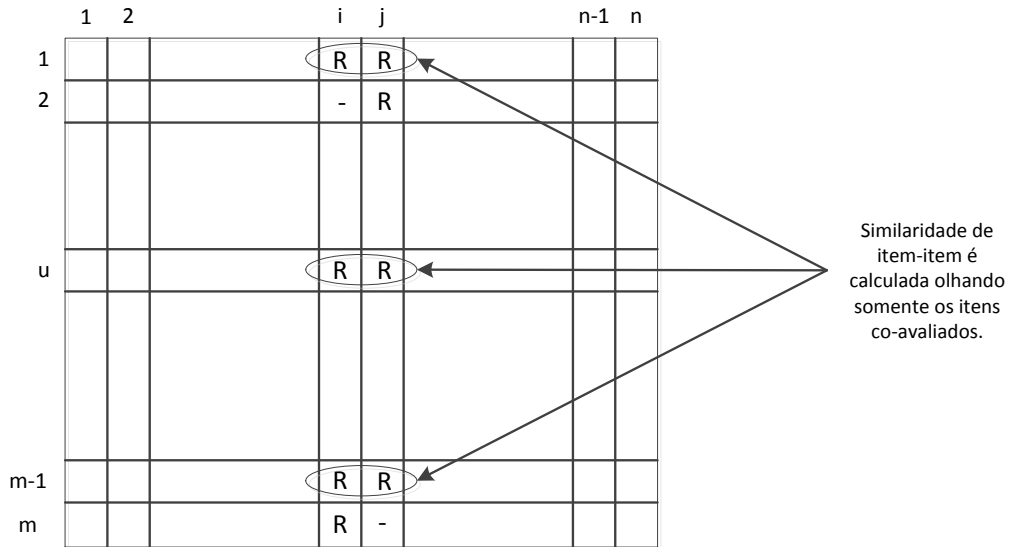


Figura 1 - Exemplo de itens que foram avaliados em comum para o cálculo da similaridade. (SARWAR *et al.*, 2001)

A correlação utilizando o cosseno, também muito utilizada na área de busca e recuperação de informação (BRI), se baseia na representação de dois objetos na forma de dois vetores (x_a e x_b) em um espaço e a similaridade entre os dois seria dada pelo cosseno do ângulo formado entre eles.

$$\cos(x_a, x_b) = \frac{x_a^T x_b}{\|x_a\| \|x_b\|}$$

Equação 6 - Fórmula da similaridade utilizando o cosseno.

No contexto do sistema de recomendação, esse valor pode ser utilizado para expressar a similaridade de dois usuários ou dois itens onde o espaço é formado pelas notas co-avaliadas entre os dois objetos x e y , ou seja, $S_{xy} = \{s \in S | r_{x,s} \neq \emptyset \ \& \ r_{y,s} \neq \emptyset\}$, onde S_{xy} é o conjunto de intercepção entre os dois e $sim(x, y)$ variando de $[-1 \ 1]$.

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} r_{xs} r_{ys}}{\sqrt{\sum_{s \in S_{xy}} r_{xs}^2 r_{ys}^2}}$$

Equação 7 - Similaridade utilizando a correlação através do cosseno.

O problema dessa similaridade é que a medida não leva em conta a diferença entre a média e a variância das notas feitas pelo usuário ou item x e y . Outra medida de similaridade

utilizada é a correlação de Pearson, que consegue remover os efeitos indesejáveis da similaridade por cosseno e varia entre [0 1].

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2 \sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$

Equação 8 - Similaridade utilizando a correlação de Person.

2.1.2.2.2 - Modelo

Em contraste com os sistemas baseados na vizinhança, que utilizam diretamente a notas armazenadas para realizar a predição de uma nota por um usuário u para o item i , o sistema de recomendação baseado em modelo tenta criar um modelo matemático para tal tarefa. Técnicas de aprendizado de máquina ou algoritmos de mineração de dados são normalmente utilizados para a criação de um modelo que consiga identificar os padrões complexos existentes e traduza em um modelo que simule o comportamento dos usuários.

A recomendação baseada em modelo tem sido investigada para tentar resolver os problemas decorrentes da filtragem colaborativa baseada em memória. Em (BREESE *et al.*, 1998) assume que a nota r é uma variável aleatória inteira e o objetivo é calcular as probabilidades condicionais de um usuário dar uma nota a um item e com isso a nota prevista r'_{ui} será a ponderação das probabilidades com o valor absoluto da nota. Como mostra a Equação 9.

$$r'_{ui} = E(u, i) = \sum_{r=0}^n r \times \Pr(r'_{ui} = r | r'_{ui'}, s' \in S_u)$$

Equação 9 – Modelo probabilístico.

O desafio desse método é o cálculo da probabilidade condicional. Em (BREESE *et al.*, 1998), apresenta duas técnicas diferentes, o modelo por agrupamento e a rede Bayesiana. Em (CHEN & GEORGE, 1999), um modelo Bayesiano é utilizado para agrupar os usuários em grupos que compartilham a mesma distribuição de notas. Os parâmetros do modelo são estimados utilizando uma Cadeia de Markov. Recentemente, foram propostos alguns trabalhos envolvendo métodos probabilísticos mais sofisticados, como Processo de Decisão Markoviano (PDM) (SHANI *et al.*, 2006) e Análise Semântica Latente Probabilística (ASLp) (HOFMANN, 2003, 2004).

Outra classe de técnicas são os algoritmos baseados em modelos de regressão (SU & KHOSHGOFTAAR, 2009), onde tentam aproximar a nota utilizando este tipo de modelo. Consideramos que $X = (X_1, X_2, \dots, X_u)$ são variáveis aleatórias que representa as preferências de um usuário sobre os itens. Sendo Λ é uma matriz $u \times k$, $N = (N_1, N_2, \dots, N_u)$ é uma variável aleatória que representa o ruído das escolhas dos usuários e R é $n \times i$ onde r_{ui} é a nota de um usuário u para um item i . Podemos representar o modelo de regressão na Eq.

$$R = \Lambda X + N$$

Equação 10 - Modelo de regressão.

Dentro da literatura existem diversas soluções para preencher os valores faltantes da matriz R , ou seja, realizar a predição das notas faltantes. Em (CANNY, 2002) se propôs a utilização de algoritmos de *Expectation Maximization* (EM) (DEMPSTER *et al.*, 1977) para preencher os valores faltantes da matriz.

Em (FUNK, 2012) se propôs uma simplificação do modelo de regressão e a utilização da técnica de Decomposição em Valores Singulares (SVD) para construção do modelo. Com isso, surgindo uma classe de algoritmos baseada em modelos que utiliza o conceito de variáveis latentes (DUMAIS, 2005) para a construção do modelo para recomendação.

Modelos utilizando fatores latentes são algoritmos que tentam explicar a preferência de um item para um usuário caracterizando os mesmos em fatores. Exemplificando, no caso de um sistema de recomendação de filmes podemos interpretar os valores latentes de um filme como o quanto este se caracteriza com relação a uma categoria. Já no caso do usuário seria a preferência deste sobre as categorias. Como é exemplificado na Figura 2.

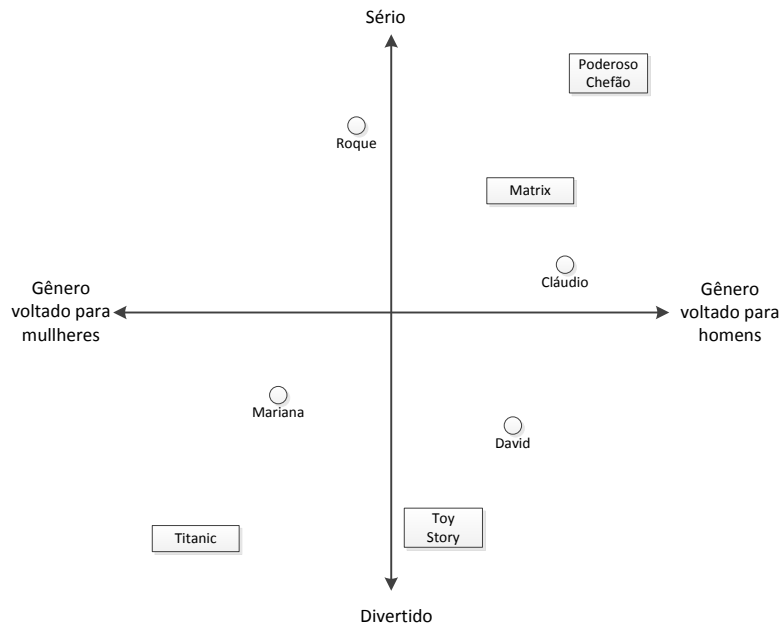


Figura 2– Ilustração simplificada da ideia das variáveis latentes onde é utilizado quatro fatores, ou melhor, categorias para caracterizar o usuário e o item (nesse caso um filme). (KOREN *et al.*, 2009)

Em (FUNK, 2012), utiliza-se a técnica de Decomposição em Valores Singulares para a inicialização do modelo onde a matriz de nota $m \times n$ é decomposta em PSQ^T , sendo que P é $m \times m$, Q é $n \times n$ e S é a matriz singular $m \times n$. Podemos manter κ colunas das matrizes P , S e Q para obter uma matriz original aproximada, como mostra a Figura 3. Podemos interpretar a matriz P como os κ fatores latentes do usuário e a matriz Q como o do item (KOREN *et al.*, 2009).

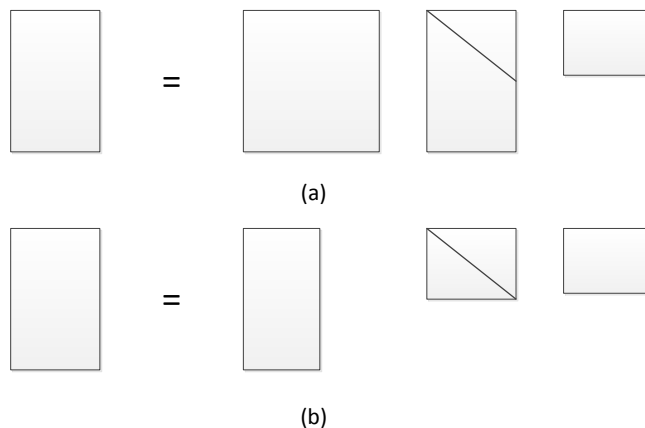


Figura 3 – Imagem acima a decomposição SVD. A imagem abaixo a decomposição aproximada.

Com a fatoração temos que para cada usuário u existe um vetor $p_u \in \mathbb{R}^k$ e para cada item i existe um vetor $q_i \in \mathbb{R}^k$ que corresponde os fatores latentes de cada um. No trabalho (FUNK, 2012) assume que a nota r_{ui} de um usuário u para um item i pode ser aproximada pela multiplicação dos fatores de cada um.

$$r_{ui} = q_i^T p_u$$

Equação 11 – Nota aproximada após a decomposição por valores singulares.

O maior desafio é que a matriz de notas é esparsa e com isso existem diversos valores faltantes na realização da decomposição. Então no trabalho utilizou-se, após a decomposição, a realização de um procedimento de aprendizado. Com o objetivo de minimizar o erro quadrático da soma, foi utilizado o algoritmo do gradiente descendente para convergir o modelo. No trabalho propôs-se a utilização da constante λ com o valor de 0.02 para a regularização da função.

$$\min_{q_i, p_u} \sum_{(u,i) \in \mathcal{X}} (r_{ui} - q_i^T p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2)$$

Equação 12 – Erro da soma quadrático utilizando o modelo sugerido por Simon Funk em (FUNK, 2012).

O algoritmo do gradiente descendente irá percorrer todas as avaliações r_{ui} do conjunto de treinamento e para cada uma irá calcular o erro em relação à nota prevista r'_{ui} . Para cada avaliação do conjunto de treinamento será feita a correção dos parâmetros q_i e p_u com relação ao erro e_{ui} . No trabalho propôs-se a utilização da constante γ com o objetivo de controlar a taxa de aprendizado e o valor utilizado foi de 0.001.

$$e_{ui} = r_{ui} - q_i^T p_u$$

$$q_i \leftarrow q_i + \gamma \times (e_{ui} p_u - \lambda q_i)$$

$$p_u \leftarrow p_u + \gamma \times (e_{ui} q_i - \lambda p_u)$$

Equação 13 – Algoritmo para a convergência do modelo proposto por Simon Funk (FUNK, 2012).

O trabalho acima foi estendido em (PATEREK, 2007, TAKÁCS *et al.*, 2008, KOREN *et al.*, 2009), onde expandiram o conceito da função de regressão. A Equação 11 demonstra que a nota de um usuário para um item é o comportamento dos seus fatores latentes. Em muito

sistemas de recomendação se nota que um item e um usuário possuem uma tendência (*bias*) independentemente de qualquer interação, ou seja, um item que possui uma nota baixa vai tender a manter essa nota para os demais usuários.

Então, o objetivo não é expressar a nota como uma simples interação dos fatores latentes do usuário pelo item ($q_i^T p_u$). Simplesmente existe uma porção da nota que já é a tendência do usuário e do item sobre a média global inerente aquele sistema de recomendação, e a outra porção é dado pelos valores latentes. Podemos dizer que uma nota r_{ui} possui uma tendência b associada tanto do item quanto do usuário em relação à média global μ .

$$b_{ui} = \mu + b_i + b_u$$

Equação 14 - Representação da tendência inerente a uma nota.

Um exemplo, utilizando um sistema de recomendação de filmes, é o usuário chamado Cláudio avaliando o filme *Matrix*. A média global do sistema μ é de 3.6, e esse filme, devido ao seu grande sucesso, possui uma tendência de receber notas 0.5 acima da média. Já o usuário Cláudio, normalmente muito crítico, possui uma tendência negativa de 0.3. Nesse exemplo a tendência relativa a esse usuário sobre esse filme é de 3.8 ($3.6 + 0.5 - 0.3$). Inserindo o conceito de tendência na Equação 11 ficamos com:

$$r_{ui} = \mu + b_i + b_u + q_i^T p_j$$

Equação 15 - Nota aproximada após a decomposição por valores singulares e utilizando o conceito de tendência.

Estendendo o modelo de aprendizado com os novos parâmetros (b_i, b_u) temos a nova função objetivo:

$$\min_{b_*, q_*, p_*} \sum_{(u,i) \in \mathcal{X}} (r_{ui} - \mu - b_i - b_u - q_i^T p_j)^2 + \lambda (b_i^2 + b_u^2 + \|p_u\|^2 + \|q_i\|^2)$$

Equação 16 – Erro da soma quadrático utilizando o conceito de tendência.

Utilizando a mesma técnica proposta por Funk em (FUNK, 2012) que utiliza o gradiente descendente para a resolução da função objetivo, temos:

$$b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u)$$

$$b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i)$$

$$p_u \leftarrow p_u + \gamma(e_{ui}q_i - \lambda p_u)$$

$$q_i \leftarrow q_i + \gamma(e_{ui}p_u - \lambda q_i)$$

Equação 17 - Algoritmo para a convergência utilizando o conceito de tendência.

Dentro da literatura existem outras extensões desse modelo (PATEREK, 2007, TAKÁCS *et al.*, 2008, RICCI *et al.*, 2011). Em todas elas estende-se o modelo adicionando novos parâmetros ou criando novos conceitos, como exemplo o tempo ou o contexto, e com isso adicionando novas informações ao modelo para que ele possa prever melhor. Essa classe de algoritmo ganhou uma grande importância em 2006 devido a uma competição feita pela empresa de aluguel de filmes online chamada *Netflix*⁷ onde o objetivo era melhorar o estado da arte nos algoritmos de recomendação. Nessa competição os melhores resultados foram realizados pelos os algoritmos de fatorização (KOREN *et al.*, 2009).

2.1.1.2 - Baseado em Conteúdo

O princípio geral da recomendação baseada em conteúdo é analisar os itens avaliados ou consumidos pelo usuário com o objetivo de identificar características comuns entre eles e assim determinar o conjunto de interesses ou perfil do usuário. Através desse perfil é possível filtrar os itens que possuem certas características que agrada o usuário e por fim recomendá-los. Esse tipo de abordagem tem como principais origens a área de Busca e Recuperação e na Inteligência Computacional. (RICCI *et al.*, 2011)

Podemos citar como exemplo uma aplicação de recomendação de filmes. Usando a abordagem baseada em conteúdo, a recomendação será dada pela identificação do padrão do comportamento do usuário verificando as descrições dos filmes melhores avaliados por este a fim de encontrar outros filmes similares para serem recomendados. Além da descrição textual

⁷ <http://www.netflix.com>

do filme poderiam ser utilizados outros atributos para serem avaliados como a lista de atores e diretores, categorias dos filmes e diversos outros.

O perfil de um usuário é uma estrutura onde se armazena os interesses dele sobre certas propriedades dos itens. Os algoritmos baseados em conteúdo exploram somente as informações implícitas e explícitas dos itens avaliados pelo usuário e essa particularidade que maior diferencia com relação à filtragem colaborativa onde os gostos dos usuários semelhantes são utilizados para a recomendação. Esse fator torna a recomendação dessa abordagem independente.

Essa característica da utilização dos gostos do usuário explícitos e implícitos para recomendação possibilita nessa abordagem indicar o motivo de cada recomendação. Durante a recomendação poderia ser listados os indicadores que levaram o item ser recomendado ao usuário gerando um aumento da confiança do usuário sobre o item recomendado. Diferentemente da filtragem colaborativa onde a motivação da recomendação é difícil de citar.

Outra distinção com relação às outras abordagens é a possibilidade de recomendar um item que nunca foi avaliado por algum usuário, ou seja, um item novo no sistema. Na filtragem colaborativa, por causa de sua filosofia, o novo item somente seria recomendado quando vários usuários tivessem avaliado ele. No caso da abordagem baseada em conteúdo bastaria o algoritmo reconhecer os descritores do item, com isso já seria o suficiente recomendá-lo para algum usuário que possui um perfil que possui interesse por essas características.

Uma limitação dessa classe de algoritmos é o número de propriedades escolhidas para serem analisadas ou as que estão explicitamente associadas aos itens. No exemplo dado anteriormente, na recomendação de filmes para usuários era feita baseada na associação do perfil do filme com o do usuário. Se conhecêssemos a idade do usuário, porém não soubéssemos o seu sexo, isso afetaria drasticamente a recomendação. Da mesma forma se a categoria do filme fosse desconhecida não seria possível prover boas recomendações. Por outro lado, saber muito sobre os itens e/ou os usuário não possui uma correlação direta com a qualidade da recomendação e afetaria o desempenho, pois existiria um enorme volume de dados para analisar.

Além disso, existe uma severa limitação que é a recomendação “mais dos mesmos”, ou seja, o usuário nunca receberá uma recomendação que diferencie do seu perfil e sempre recomendando itens similares àqueles que foram consumidos (ADOMAVICIUS & TUZHILIN, 2005). Existe na literatura abordagens que tentam evitar essa superespecialização, como exemplo, pode-se adicionar um caráter aleatório ao algoritmo, afinal, um dos objetivos dos sistemas de recomendação é surpreender o usuário. Em (SHETH, 1993), os autores propõem a utilização de algoritmos genéticos para tornar as recomendações menos homogêneas.

2.1.1.3 - Abordagens Híbridas

Os algoritmos híbridos possuem como característica a utilização de diversos algoritmos inclusive de abordagens distintas com o objetivo de realçar o potencial de cada um e compensar as suas deficiências. De modo geral, essa união é dada pela criação de uma técnica mesclando os conceitos de outros algoritmos da literatura ou utilizando diversos algoritmos combinando o seu resultado (ADOMAVICIUS & TUZHILIN, 2005). Dentro da literatura existem quatro principais abordagens para realizar a combinação.

O primeiro consiste na criação em um algoritmo de troca entre técnicas de recomendações utilizando algum critério. Cada técnica seria independente e seria utilizada uma combinação dos resultados, como exemplo a média ponderada dos resultados individuais, ou um critério de votação para a seleção do algoritmo utilizado (CLAYPOOL *et al.*, 1999), como exemplo o esquema de votação (PAZZANI, 1999).

Outro trabalho que utiliza a troca entre técnicas de recomendação é o P-Tango (CLAYPOOL *et al.*, 1999). Esse sistema utiliza pesos para controlar a dinâmica da recomendação. Ele aumenta ou diminui dependendo do aumento do número de avaliações feitas por alguma técnica. Outra abordagem é a utilização da escolha do melhor resultado considerando alguma medida de qualidade estipulada (BILLSUS & PAZZANI, 2000, TRAN & COHEN, 2000).

A segunda proposta é a mesclar as técnicas de recomendação principalmente com o objetivo de reduzir as deficiências dos algoritmos utilizando conceitos dos outros. No trabalho (BALABANOVIĆ & SHOHAM, 1997, PAZZANI, 1999) utilizam a filtragem colaborativa com abordagem de memória para realizar a recomendação. Contudo, a similaridade entre os

usuários era feita utilizando os seus perfis. Essa tática tem como objetivo remover o efeito da esparsidade dos dados e o problema quando um novo usuário entra no sistema, pois o seu perfil sempre será preenchido no cadastrado.

Em (MELVILLE *et al.*, 2002), propõe-se um método chamado “Filtragem Colaborativa Impulsionada por Conteúdo”, onde as informações sobre o conteúdo são utilizadas juntamente com as avaliações feitas pelo usuário sobre o item para criar uma matriz completa. Por fim, é utilizado a filtragem colaborativa utilizando abordagem por memória para realizar a recomendação.

Outra abordagem, inverso da anterior, utiliza os aspectos colaborativos nos algoritmos baseados em conteúdo. Dentro dessa categoria normalmente são utilizadas técnicas de redução de dimensionalidade sobre um grupo de perfis de usuários. Em (SOBOROFF & NICHOLAS, 1999), é utilizado a Indexação Semântica Latente (ISL) (DEERWESTER *et al.*, 1990) para criar uma visão colaborativa dos perfis do usuário. Esses perfis são representados em um novo espaço onde as similaridades entre eles poderiam ser exploradas mais eficientemente. Essa mudança resultaria em um melhoramento no desempenho em comparado com a recomendação utilizando a abordagem por conteúdo pura.

Por último é a construção de modelos que contemplam tanto os aspectos de conteúdo e colaborativo. Através de um modelo global onde envolva as informações das notas e das informações dos usuários e dos itens é proposto em (ANSARI *et al.*, 2000, CONDLIFF *et al.*, 1999). Nesse trabalho, o modelo gerado utiliza a análise probabilística Bayesiana, onde os parâmetros são estimados pelo método Cadeia de Markov.

2.2 – Aprendizado de Máquina

A partir da década de 70, houve o surgimento das novas tecnologias de informação e de comunicação (*NTICs*) devido à terceira revolução industrial. Essas tecnologias foram capazes de armazenar uma grande quantidade de dados que nunca a humanidade presenciou até então. No dia a dia estamos cercados por computadores como bancos, hospitais, laboratórios científicos e outros que vem armazenando dados incessantemente.

O maior desafio com essa quantidade de informação é utilizar essas com o objetivo de ajudar as pessoas no dia a dia. Como exemplo, seria possível o banco com as informações da

movimentação do cliente detectar uma fraude rapidamente? Se hospitais com todas as informações sobre a pessoa doente consegue automaticamente definir melhores tratamentos para esta? Em uma locadora, com todas as informações de aluguel de filmes do cliente, conseguir recomendar o próximo filme para este?

O ser humano é capaz de realizar essa tarefa facilmente dependendo do tamanho e da complexidade das informações. Na Figura 4, podemos comparar os dados na tabela e seu respectivo gráfico. É fácil de visualizar o padrão das classes no gráfico. No caso do ser humano o problema, principalmente, é o aumento do número de dimensões tornando impossível esse trabalho e no mundo real as informações possuem uma grande quantidade de dimensões.

x_1	x_2	Class
0.1	1	1
0.15	2	2
0.48	0.6	3
0.1	0.6	1
0.2	0.15	2
0.5	0.55	3
0.2	1	1
0.3	0.25	2
0.52	0.6	3
0.3	0.6	1
0.4	0.2	2
0.52	0.5	3

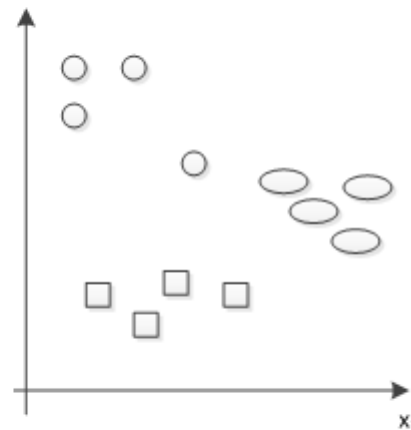


Figura 4 – Um conjunto de dados e os pontos plotados no gráfico. (MARSLAND, 2009)

Devido a essa limitação do ser humano e o desejo de extrair informações e padrões complexos dos dados para poder tomar uma decisão inteligente, a área de Aprendizado de Máquina tornou-se popular. Por causa dessas características essa área é multidisciplinar. Ela

engloba a inteligência artificial, probabilidades e estatística, filosofia, psicologia, neurobiologia, teoria de controle e outras áreas. (MARS LAND, 2009)

O Aprendizado de Máquina tem como objetivo tomar uma decisão através do aprendizado e da extração de padrões de um conjunto de dados. Esse conjunto é chamado pela literatura de conjunto de treinamento. Esse aprendizado é a busca no espaço de hipóteses pela hipótese H_x que maximize uma função de ganho. (MITCHELL, 1997a)

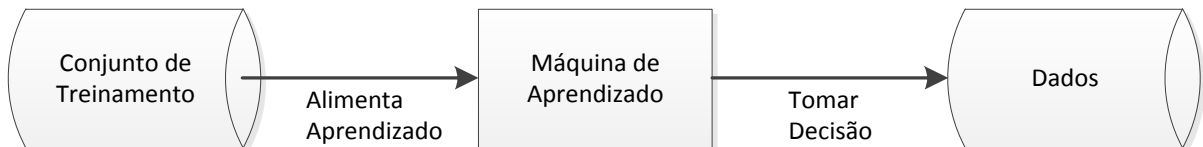


Figura 5 - Processo de Aprendizagem – (Própria Autoria)

(MITCHELL, 1997b) define Aprendizado de Máquina como:

Um programa de computador é dito aprender a partir da experiência E com respeito a uma classe de tarefas T e medir o desempenho P , se o seu desempenho em tarefas em T , tal como medido por P , melhora com a experiência E .

Por exemplo, um site de aluguel de filmes online utiliza um algoritmo que é capaz de aprender os gostos dos seus usuários e através desse aprendizado consegue recomendar filmes que os seus usuários gostem. (MITCHELL, 1997b) através de sua definição de Aprendizado de Máquina, especifica o conceito através de três características: a tarefa, a medida desempenho e a experiência de aprendizado. Separando o exemplo utilizando essas três características, temos:

- Tarefa T : Recomendar filmes;
- Medida de Desempenho P : O erro entre a nota dada após o usuário assistir o filme e a nota dada pelo algoritmo;
- Experiência de Aprendizado E : Aprendendo com as notas dadas pelos usuários pelo os filmes.

Aprendizado de Máquina é utilizado quando não existe um algoritmo que consiga resolver o problema em questão e normalmente o problema não possui uma resposta certa. O

que ocorre é que existe um valor para informar o quanto aquela resposta é certa com o objetivo de fazer o algoritmo definir uma nova hipótese na qual melhore essa resposta. Com isso o algoritmo deve obter uma hipótese que tente generalizar o problema utilizando o conjunto de treinamento. Os algoritmos de Aprendizado de Máquina podem ser agrupados em diversas categorias dependendo de como eles respondem essas perguntas. As quatro principais classes são: (MITCHELL, 1997b)

- **Aprendizado Supervisionado:** Esse tipo de aprendizado tem como objetivo utilizar os exemplos dados com suas respectivas respostas e generalizar para dar as respostas corretas a um conjunto de dados.
- **Aprendizado Não Supervisionado:** Neste caso, não existe resposta certa para o conjunto de dados. O objetivo é tentar agrupar os dados para categorizar estes utilizando as similaridades entre eles.
- **Aprendizado por Reforço:** Esse aprendizado pode ser considerado entre o aprendizado supervisionado e não supervisionado. Como no primeiro, o algoritmo recebe uma resposta quando o resultado é errado, mas não possui a informação de quanto à resposta é certa.
- **Aprendizado Evolutivo:** Nele o aprendizado tem como consequência a adaptação do modelo com relação ao tempo não possuindo valor certo ou errado e sim aquele que maximize a função objetivo de desempenho.

Na próxima subseção iremos discutir a classe de algoritmos de aprendizado supervisionado. Essa classe será utilizada neste presente trabalho.

2.2.1 – Aprendizado Supervisionado

Aprendizado supervisionado é uma classe de algoritmos dentro do aprendizado de máquina que possui a característica de inferir uma função dado um conjunto de dados que é chamado de conjunto de treinamento. Esse conjunto é dado pelo par entrada e saída onde a entrada possui as características do exemplo em si e a saída o valor desejado. Através deste conjunto de dados o algoritmo irá se corrigindo e adequando a sua função com o valor esperado de saída.

O processo de aprendizado é dado pelo mapeamento da saída em relação à entrada através da interação contínua com o ambiente. Através de um agente, que chamaremos de professor, para cada resposta dada a uma instância do ambiente esse agente informará ao algoritmo se a resposta dada está certa ou errada. Através dessa resposta será avaliado o erro quando a resposta dada for avaliada como errada para ser utilizado na correção do mapeamento. Essa interação é finalizada quando é minimizado um índice escalar de desempenho.

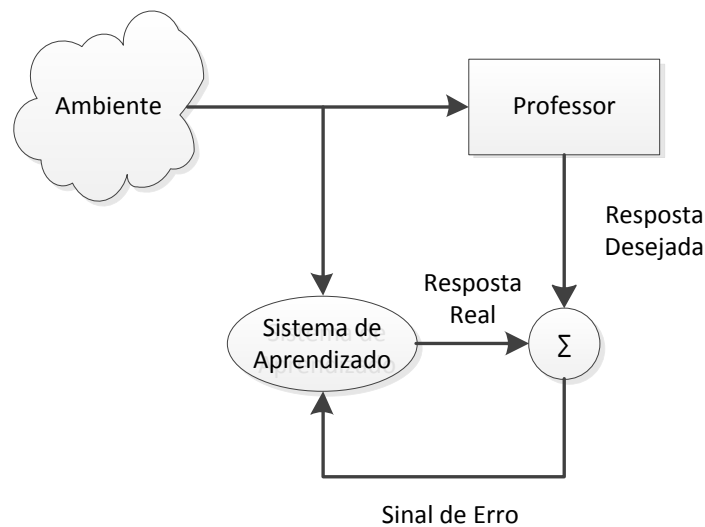


Figura 6 – Processo de Aprendizado Supervisionado.

Esse paradigma de aprendizado pode ser comparado com a situação de uma sala de aula. Existem diversos conhecimentos que os alunos devem aprender com suas descrições e resposta. O professor possui a tarefa de verificar a resposta de cada aluno e informar para cada uma se ela está correta e caso contrário será informado o motivo. Esse procedimento é realizado até o momento que os alunos consigam aprender o conteúdo.

(BISHOP, 2006) descreveu com uma maior formalidade o problema de aprendizado supervisionado: dado um conjunto de treinamento na forma $\{(x_1, y_1), \dots, (x_n, y_n)\}$, o algoritmo de aprendizado procura no espaço de hipóteses G a função que faça o mapeamento do espaço de entrada X em relação à da saída Y , ou seja, $g: X \rightarrow Y$. Podemos representar a função g utilizando a função de erro $f: X \times Y \rightarrow \mathbb{R}$, onde a função g é definida pelo valor que maximize a função de erro: $g(x) = \arg \max f(x, y)$. Dado F o espaço das funções dos erros.

Na literatura existem diversos algoritmos de aprendizado supervisionado e suas diferenças são dadas na modelagem do espaço de hipóteses G e do erro F , e conseqüentemente na forma de procurar a função que maximize a função de erro $f(x, y)$. Embora G e F podem ser qualquer espaço de função, existem algoritmos que utilizam modelos probabilísticos para definirem os dois espaços, onde g é definido pela probabilidade condicional $g(x) = P(y|x)$ ou a função f na forma de probabilidade $f(x, y) = P(x, y)$. Como exemplo desses modelos são os algoritmos Naive Bayes e a Regressão Logística.

O aprendizado supervisionado pode ser classificado com relação ao conjunto de saída Y caso os seus valores sejam discretos ($Y \in \mathbb{N}$) ou contínuos ($Y \in \mathbb{R}$) denominados respectivamente classificação e regressão. O problema utilizando os valores discretos pode ser modelado como um problema contínuo como o conjunto do $\mathbb{N} \subset \mathbb{R}$, mas o contrário não é possível já que o conjunto dos naturais consegue expressar todos os valores do outro.

O problema de classificação consiste em, dado um conjunto de N classes possíveis e um vetor de dados de entrada, decidir para cada entrada a que classe ele pertence utilizando como base um conjunto de exemplos para cada uma das classes. Uma característica inerente ao problema é que uma entrada pertence exclusivamente e somente uma das N classes definidas pelo contexto. Como exemplo, dado um conjunto de características físicas de uma pessoa, queremos classificar cada uma em dois rótulos: “homem” ou “mulher”. Uma pessoa é necessariamente classificada em um dos rótulos e somente um deles.

A essência desse tipo de problema é tentar encontrar no espaço de características as fronteiras de decisão onde seria possível separar os dados em diferentes classes. Na Figura 7 utilizamos como exemplo um conjunto de dados utilizando duas dimensões de características e cada dado pode pertencer a três distintas classes. Nesse exemplo é exemplificada a fronteira de decisão sendo que no gráfico da esquerda utilizando retas e do lado direito curvas. Nesse caso não é possível separar as classes linearmente.

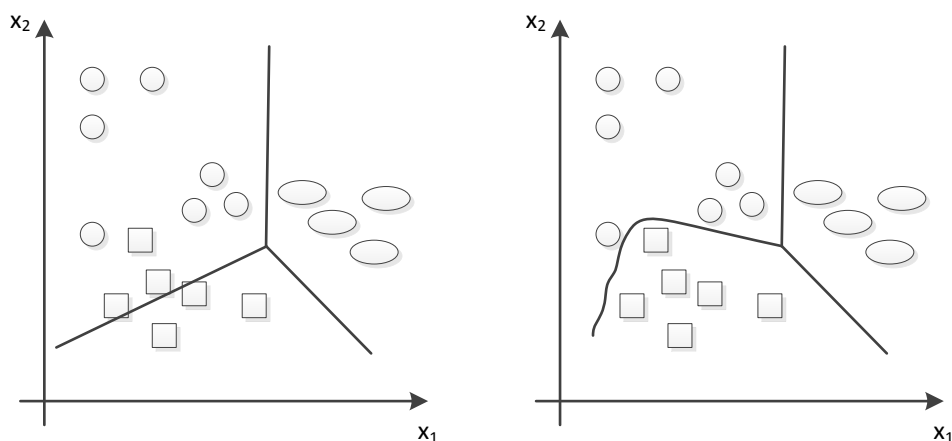


Figura 7 - Dois gráficos exemplificando a fronteira de decisão. No lado esquerdo utilizando retas e no lado direito utilizando curvas.

O problema de regressão consiste em interpolar uma função que melhor se adeque aos dados com o objetivo de prever o valor em um determinado ponto. Como exemplo, supondo os dados na Tabela 3 onde a coluna x é o tamanho em m^2 de uma casa (conjunto de entrada) e t é o valor em dólar dessa residência. Supondo que uma nova casa quer ser avaliada e ela possui $280 m^2$. Note que esse valor não possui nenhum exemplo. Com isso o objetivo da regressão é encontrar uma função que melhor interpole esses dados para prever o novo dado.

Tabela 3 – Valor de uma residência por seu respectivo tamanho.

$x (m^2)$	$t (\$)$
100	150.000
200	300.000
400	450.000
600	1.000.000
800	2.000.000

Na próxima subseção serão apresentados em linhas gerais os algoritmos de aprendizado de máquina supervisionado que serão utilizadas neste presente trabalho, como Redes Neurais, aprendizado Bayesiano e outros.

2.2.1.1 – Algoritmos

2.2.1.1.1 – Aprendizado Bayesiano

Em aprendizado de máquina, o objetivo é encontrar o melhor espaço de hipóteses H tendo o conjunto de treinamento D como base. Podemos definir que o melhor espaço H é aquele que define a hipótese mais provável, dado a probabilidade a priori do conjunto de dados D em relação a diversas hipóteses H . O teorema de Bayes é um método que é capaz de calcular diretamente essas probabilidades. (MITCHELL, 1997b)

Definindo que $P(h)$ é a probabilidade de uma hipótese acontecer no conjunto de dados. $P(D)$ é a probabilidade do conjunto de dados ser observado. $P(D|h)$ é a probabilidade de um dado ser observado dado uma hipótese h e por último $P(h|D)$ é a probabilidade de uma hipótese h acontecer dado um conjunto de dados observados. Esse valor é o que o problema de aprendizado de máquina interessado em inferir. Com essas definições podemos aplicar o teorema de Bayes:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Equação 18 - Teorema de Bayes.

Para ficar melhor claro a conexão do teorema com os problemas de aprendizado de máquina iremos detalhar um exemplo típico. Em um determinado país existem dois tipos de laranjas e nesse país existem os estados A e B. As laranjas que são cultivadas no estado A são nomeadas de laranjas de A, e as que são cultivadas no estado B, são chamadas de laranjas de B. Os dois tipos são distintos, porém são muito parecidos, sendo quase impossível de distinguir apenas pelo olhar. As colheitas são transportadas para a capital, onde são misturadas e vendidas para todo o país.

Supondo que um observador, ao ver as laranjas sendo retiradas aleatoriamente de um saco com ambos os tipos, tenha que prever qual o tipo da próxima laranja a ser retirada. Iremos considerar as laranjas do estado A como sendo da classe C_a e as laranjas de B como sendo da classe C_b . Para esse trabalho o observador utilizará o peso w da laranja para tentar diferenciar cada uma com o objetivo de tentar prever de qual estado ela pertence.

Como laranjas diferentes resultarão em pesos diferentes, assim podemos assumir que w é uma variável aleatória contínua cuja distribuição depende de uma variável discreta (classe) C_i . Portanto, a função densidade de probabilidade condicional é denotada por $P(w|C_i)$. Supondo que o observador possui o conhecimento da probabilidade de uma classe sair ($P(C_i)$) e de posse de w e $P(w|C_i)$ podemos calcular a probabilidade posteriori definida por $P(C_i|w)$ através da fórmula de Bayes da Equação 18:

$$P(C_i|w) = \frac{P(w|C_i)P(C_i)}{P(w)}$$

Equação 19 – Aplicação do teorema de Bayes no exemplo dado.

A probabilidade denotada por $P(C_i|w)$ é a probabilidade de o elemento pertencer à classe C_i , dado que o peso ω é conhecido. Caso $P(C_1|w) > P(C_2|w)$ pode-se concluir que o dado pertence à classe C_1 , caso contrário à classe C_2 . Uma observação importante é que $P(w)$ é utilizada para normalizar a função para que ela fique entre [0 1]. Tendo isso em mente podemos ignorar e considerar somente o maior valor de $P(C_i|w)$.

Caso o conjunto D possua mais de uma característica conhecida, ou seja, um vetor de variáveis D pertence a \mathbb{R}^m onde $D = (d_1, d_2, \dots, d_m)$. Com isso o fator $P(D|h)$ pode ser difícil de calcular se o espaço m for muito grande. Para tal o algoritmo supõe que as variáveis d_m são independentes. Logo $P(D|h)$ pode ser calculado da seguinte forma:

$$P(D|h) = \prod_{j=1}^m P(d_j|h) \times P(h)$$

Equação 20 – Classificador utilizando o teorema de Bayes.

Como essa suposição não é verdadeira, pois ela assume uma propriedade que nem sempre é verdadeira para facilitação dos cálculos. Em (RISH, 2001) os autores demonstram a utilização desse classificador comparando o desempenho dele supondo as variáveis independentes e dependentes. Nos dois casos ele obteve bons resultados e foi possível encontrar todas as variáveis a partir de um grupo conhecido.

2.2.1.1.2 – Árvore de Decisão

O classificador utilizando árvore decisão é uma técnica simples que utiliza regras SE-ENTÃO com o objetivo de classificar os dados de entrada em valores discretos. Mesmo devido à sua simplicidade esse algoritmo é muito utilizado devido ao seu sucesso na aplicação em alguns problemas como diagnósticos médicos e avaliação de risco em aplicação de investimentos. Outra característica é que nessa técnica é possível verificar facilmente o motivo da classificação utilizando a visualização de árvore. (MITCHELL, 1997b)

A árvore de decisão é representada por uma árvore onde cada folha é definida por uma classe do problema e os nós não terminais é dado por uma regra que possui uma condição sobre os atributos do dado com o objetivo de separar os registros em características diferentes. (MITCHELL, 1997b) Utilizando como exemplo a Figura 8 onde queremos descobrir se um usuário gostaria de assistir a um determinado filme com base aos dados do filme. No exemplo se o filme for acima do ano de 2005 o usuário gostaria de assistir o filme. Caso seja abaixo de 2005 o usuário só irá assistir se o filme for de comédia.

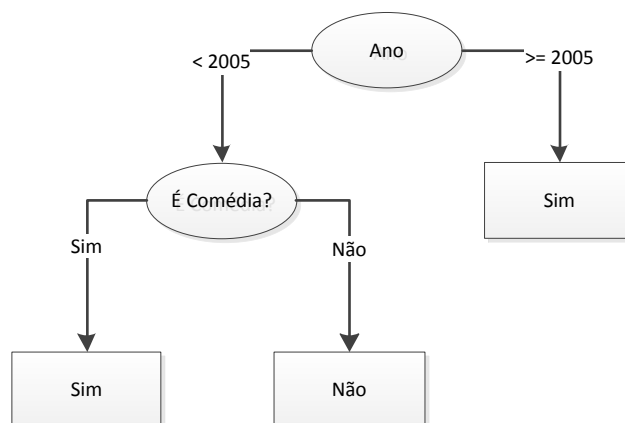


Figura 8 - Exemplo de uma Árvore de Decisão.

Em geral, a árvore de decisão é representada através de disjunções de conjunções de restrições sobre os atributos das instâncias. Cada caminho de um nó para outro representa uma restrição de um teste de um ou um conjunto de atributos. Com isso a árvore representa a disjunções dessas conjunções. Utilizando como exemplo a Figura 8 podemos representá-la como:

$$(Ver\ o\ Filme) = (Ano \geq 2005) \vee (Ano < 2005 \wedge (É\ Comédia))$$

Equação 21 – Exemplo de regras de uma árvore de decisão.

Devido a essas características podem existir milhares de possíveis árvores de decisões que podem ser construída através do conjunto de atributos e esse número possui uma relação exponencial com o tamanho do conjunto. A busca pela árvore ótima é computacionalmente inviável pelo tamanho do espaço de pesquisa. Foram desenvolvidos algoritmos que utilizam heurísticas para tentar induzir uma árvore de decisão razoavelmente precisa em uma quantidade razoável de tempo.

Um dos algoritmos mais famosos para realizar essa construção é o algoritmo de Hunt (HUNT *et al.*, 1966) que é a base dos algoritmos existentes como o ID3 (QUINLAN, 1986), o C4.5 (QUINLAN, 1993) e CART (BREIMAN, 1984). O processo de construção consiste em particionar recursivamente o conjunto de dados definindo um atributo para divisão até que cada subconjunto possua uma única classe. Existem diversos métodos para selecionar o melhor atributo para a divisão. Todas elas tendem otimizar alguma métrica, como por exemplo a entropia ou erro de classificação.

2.2.1.1.3 – Florestas Aleatórias

O modelo florestas aleatórias é um modelo que aplica as técnicas de aprendizado por agrupamento (*assemble learning*) com as árvores de decisões. O algoritmo foi introduzido por (HO, 1995) e ela é a combinação das previsões de diversas árvores de decisões onde cada árvore foi gerada através de um conjunto independente de vetores aleatórios, como demonstrado na Figura 9. Sendo que a seleção do conjunto de atributos para cada árvore é dada utilizando aleatoriedade, *boosting* ou *bagging*.

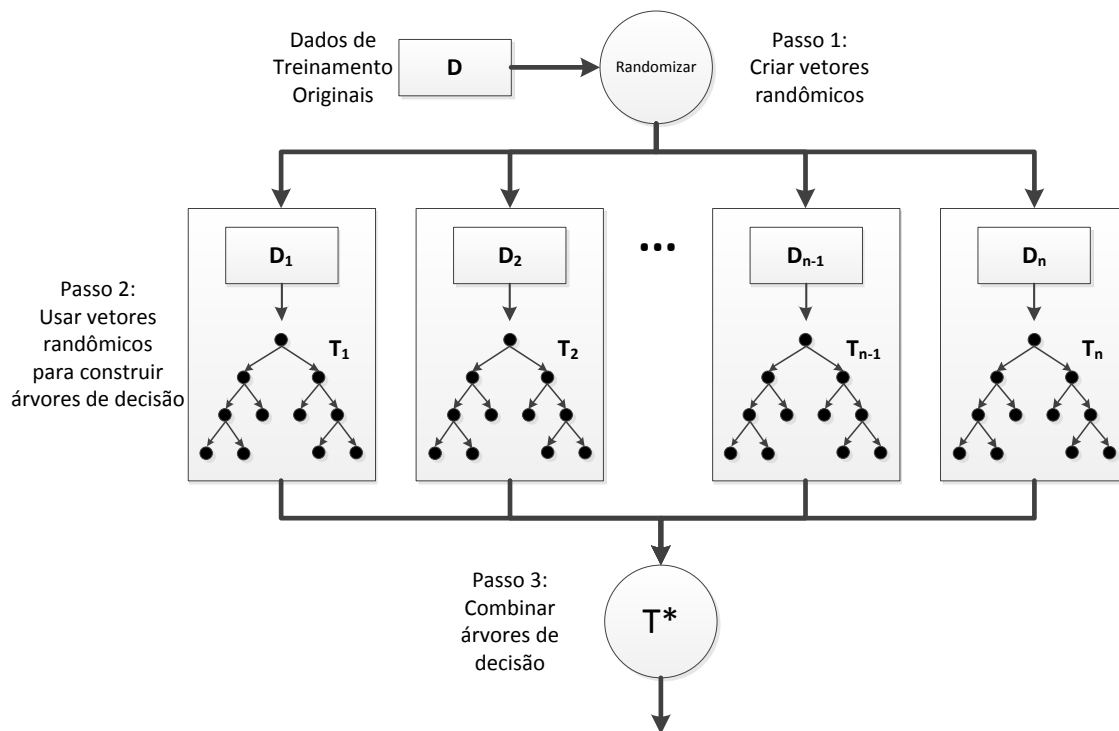


Figura 9 - Florestas Aleatórias. (TAN *et al.*, 2005)

Foi demonstrado que a floresta aleatória é um dos algoritmos mais precisos dentro da área de aprendizado de máquina, oferecendo a possibilidade de paralelização, capacidade de escalar e consegue lidar com dados de alta dimensionalidade (CARUANA & KARAMPATZIAKIS, 2008). Contudo, devido à sua complexidade e ao contrário das árvores de decisões, o algoritmo torna-se difícil à interpretação humana do resultado.

2.2.1.1.4 – Vizinho mais Próximo (K-NN)

O classificador dos k vizinhos mais próximos (*k-nearest neighbor*, *K-NN*) é um método que consiste atribuir uma classe ao elemento desconhecido usando as informações dos vizinhos mais próximos. Para tal, é necessário definir uma função de similaridade entre os pares de observações para poder definir a vizinhança de um elemento. Esse algoritmo é o algoritmo mais simples da área do aprendizado de máquina e normalmente a função definida é a distância euclidiana.

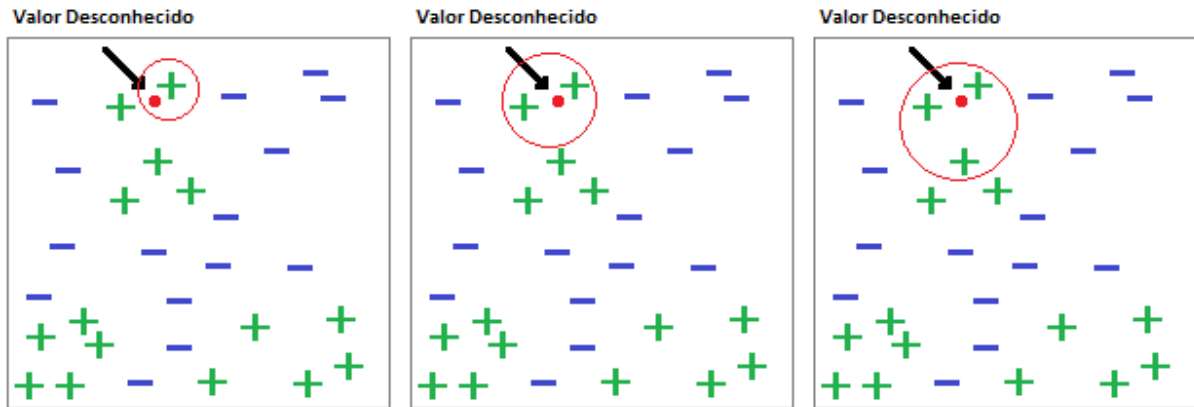


Figura 10 - Exemplo do K-NN com k valendo 1, 2 e 3 respectivamente.

Diferentemente da árvore de decisão, que define um modelo que mapeia todos os atributos do conjunto de treinamento para o rótulo da classe, o vizinho mais próximo possui uma estratégia oposta onde o processo da criação do modelo é atrasado até que eles sejam necessários para classificar os exemplos de testes. Esse tipo de abordagem é conhecido como aprendizado baseado na instância e sua maior vantagem é que o modelo gerado é definido localmente para cada instância, com isso lidando melhor com mudanças do problema.

Utilizando a distância euclidiana como exemplo, sejam $x_1 = (x_{11}, x_{12}, \dots, x_{1p})$ e $x_2 = (x_{21}, x_{22}, \dots, x_{2p})$ duas observações p -dimensionais e W o conjunto de exemplos de treinamento, a distância entre elas é dada por:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^p (x_{1i} - x_{2i})^2}$$

Para uma nova observação z , será realizado o cálculo da distância de todos os elementos de W , ou seja, $\forall w \in W, d(z, w)$. Sendo que a lista W_z de k elementos mais próximos serão aqueles que venham a possuir o maior valor de similaridade, mas nesse caso o inverso, devido à característica inversa da distância. Por fim, falta definir a estratégia para definir a classe do elemento z . Uma estratégia comum é a utilização da votação majoritária que possui o objetivo de definir o rótulo ao elemento desconhecido utilizando a classe majoritária dos seus vizinhos mais próximos:

Na abordagem da votação majoritária, cada vizinho possui o mesmo impacto sobre a classificação. Devido a essa característica o algoritmo torna-se sensível à escolha de k

conforme é demonstrado na Figura 11. Se k for pequeno, então o classificador pode ser susceptível a *overfitting* por causa do ruído nos dados. Caso o k for grande, o classificador pode inserir informação desnecessária e pode levar a classificar erroneamente. Uma forma de reduzir esse efeito é utilizando a similaridade como peso.

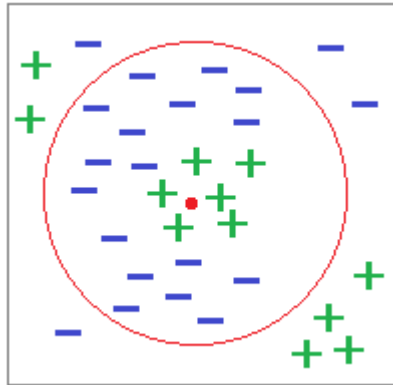


Figura 11 - - Exemplo do K-NN sendo que o k grande.

2.2.1.1.5 – Redes Neurais Artificiais (ANN)

O modelo conhecido como redes neurais artificiais é um sistema paralelo e distribuído composto por unidades de processamento simples (neurônio artificiais) que calculam determinadas funções matemáticas (normalmente não lineares). Esse modelo teve como origem o estudo em (MCCULLOCH & PITTS, 1943) onde se tentou representar e modelar eventos do sistema nervoso e culminou em (ROSENBLATT, 1958) que concentrou-se em descrever um modelo artificial de um neurônio e suas capacidades computacionais.

O cérebro humano possui em torno de 10^{10} células nervosas chamadas neurônios. Os neurônios possuem diversas ligações entre eles e ela é chamada de axônio. Os axônios são usados para transmitir impulsos nervosos entre os neurônios sempre que os neurônios forem estimulados. Um neurônio é ligado a um axônio através de dendritos, que são extensões do corpo celular. Por fim o ponto de contato entre um dendrito e um axônio é chamado de sinapse. Foi descoberto que o cérebro humano aprende através da força de conexão da sinapse entre neurônios em repetidos estímulos pelo mesmo impulso. Essa estrutura é demonstrada na Figura 12.

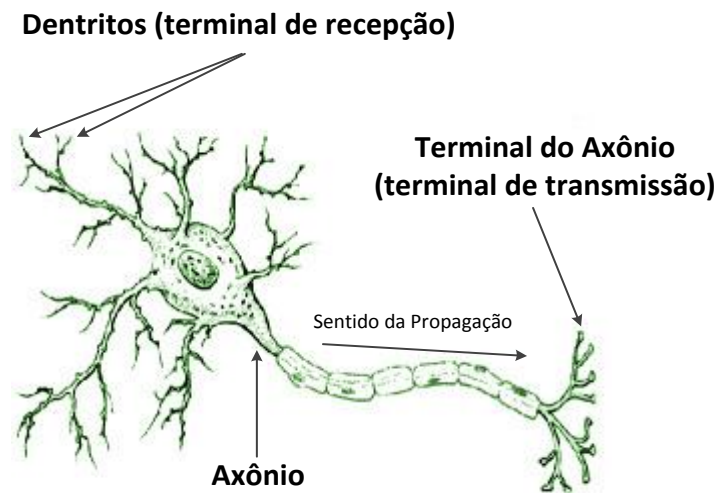


Figura 12 - Componentes do neurônio biológico⁸.

Análoga à estrutura do cérebro humano, um ANN é composto de um conjunto de unidades de processamentos interconectados e suas ligações são direcionadas. Uma unidade de processamento é chamada de perceptron. Ele é um modelo que possui n terminais de entradas (dendritos) que recebem os valores x_1, x_2, \dots, x_n (que representam os valores da ativação dos outros neurônios) e apenas um terminal de saída y (representando os axônios). Para representar as sinapses, cada terminal de entrada possui um peso associado w_1, w_2, \dots, w_n cujos valores podem ser positivos ou negativos com isso representando se a ligação é inibidora ou excitadora. Por fim, o núcleo receberá a soma dos valores $x_i w_i$ e o valor da saída será dada por uma função. Uma descrição do modelo está ilustrada na Figura 13.

⁸ Figura feita a partir da original de AJ Cann como vista em <http://www.flickr.com/photos/ajc1/8026286228/> no dia 10/10/2012, distribuída sob a licença CC-BY-SA-2.0.

x_1	x_2	x_3	y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	0	-1
0	1	1	-1
0	1	0	1
0	0	0	-1

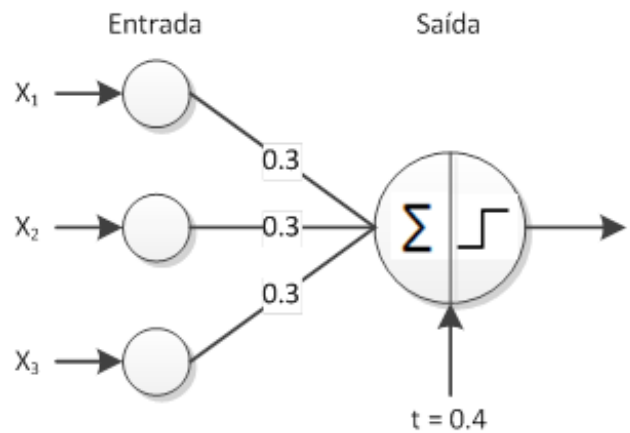


Figura 13 - Modelando uma função booleana usando um perceptron. (TAN *et al.*, 2005)

A capacidade computacional de um neurônio é limitada. Contudo, um conjunto de neurônios artificiais interligados em forma de uma rede é capaz de resolver problemas de alta complexidade. A estrutura utilizando duas camadas (escondida e de saída) é a mais utilizada devido ao seu poder computacional e universalidade na aproximação de funções contínuas, sendo que com duas camadas intermediárias podem gerar qualquer função, seja ele linearmente separável ou não (CYBENKO, 1989). Na Figura 14 demonstra um exemplo de uma rede neural utilizando multicamadas.

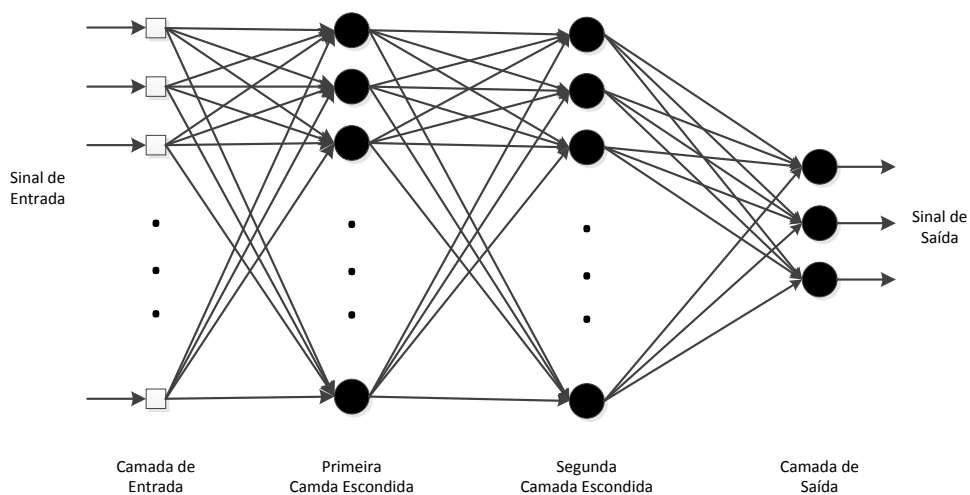


Figura 14 - Arquitetura de uma rede de perceptron utilizando duas camadas.

A definição da quantidade de neurônios em cada uma das camadas é de extrema importância, pois nela constam seu desempenho e a sua capacidade de generalização. Existem inúmeras abordagens que tentam estimar o número de neurônios na literatura, mas nenhuma delas consegue formalizar uma resposta geral para o projeto de uma rede neural.

O algoritmo mais popular para o treinamento é o *back-propagation* que, por ser supervisionado, utiliza a saída esperada e a fornecida com relação a uma entrada dos dados para realizar correções, ou seja, ajustar os pesos da rede. Basicamente, o algoritmo calcula o gradiente do erro da rede e sempre é utilizado em conjunto com um algoritmo gradiente descendente para encontrar os pesos que minimizam o erro da saída.

2.3 – Trabalhos Relacionados

A filtragem colaborativa vem sendo largamente usada pelos sistemas de recomendação devido a sua simplicidade de tratar ao problema e seus bons resultados (LINDEN *et al.*, 2003). Os métodos dessa categoria são técnicas clássicas de classificação e regressão, mas com diversas modificações para lidar com os problemas do domínio da recomendação, como por exemplo, a matriz esparsa de notas. Um exemplo de uma técnica de classificação é a dos vizinhos mais próximos na qual a similaridade entre dois usuários é calculada utilizando os itens co-avaliados. (ADOMAVICIUS & TUZHILIN, 2005)

O grande desafio na aplicação dos algoritmos de aprendizado supervisionado no domínio de um sistema de recomendação é a impossibilidade de definir diretamente um conjunto de características utilizando somente as notas dos usuários sobre os itens. Caso utilizasse diretamente esses dados como características elas iriam variar o seu tamanho com o tempo quando um usuário ou item novo entrasse no sistema. Consequentemente essas mudanças iriam alterar o modelo de aprendizado de máquina e isso invalidaria os modelos anteriores.

Em (HSU *et al.*, 2007, O'MAHONY & SMYTH, 2010, 2009, CUNNINGHAM & SMYTH, 2010), utilizam o conhecimento sobre o domínio e definem diversas métricas para a geração desse conjunto, como exemplo, a média de notas do usuário e do item, a quantidade de itens avaliados pelo usuário e outras medidas. O problema dessa abordagem é a dificuldade na seleção das melhores métricas que melhor generalizem e ajudem a treinar a máquina de

aprendizado. Inclusive em todos os trabalhos adotam utilizar outras características, além das métricas sobre as notas, para dar mais informação ao classificador.

Em (BILLSUS, 1998), o autor propõe um método independente de domínio com o objetivo de prever as notas. Ele sugere a utilização de duas novas matrizes. A primeira matriz contém a informação, se um usuário gostou do item e na segunda se ele não gostou. Caso a matriz original possua um conjunto de preferência com mais de dois elementos, ele sugere a utilização de um pivô. Esse valor poderia ser a média das notas, como exemplo, e se o valor da nota for maior o usuário teria gostado e caso contrário teria não gostado.

Como cada matriz define se é verdade ou não, se o usuário gostou do item ou não gostou. Logo, os itens que tiverem sem o valor da preferência do usuário explicitado serão considerados como o valor booleano falso nas duas matrizes. Devido ao domínio de recomendação possuir um grande número de dimensões, aproximar uma função de aprendizado utilizando essa grande quantidade de pontos, fenômeno comum chamado “maldição da dimensionalidade” (BELLMAN, 1961), torna o aprendizado inviável. Para tal, foi utilizado técnicas de redução de dimensionalidade para resolver esse problema.

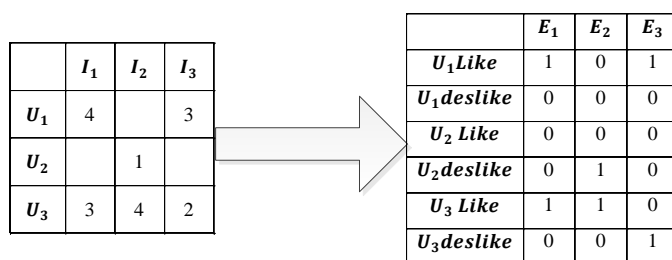


Figura 15 - Exemplo de conversão da matriz de preferências para a matriz de gosto utilizado por (BILLSUS, 1998).

Para cada usuário, um modelo é induzido com base na união das duas matrizes. Mesmo a proposta obtendo bons resultados, ele não trabalha para problemas multiclases e não é escalável em relação ao número de usuário devido a grande quantidade de modelos que devemos induzir. Esse método utiliza SVD para a redução de dimensionalidade e construir um novo espaço de características para treinar os modelos.

Devido às características intrínsecas do contexto lidado pelos sistemas de recomendação, esse tipo de problema não pode ser lidado como típico problema de aprendizado de máquina. Como consequência não é possível aplicar diretamente as diversas

técnicas da área sem a realização de modificações ou adaptações para se adequar ao problema. Seria de grande motivação se fosse possível de alguma forma criar uma arquitetura capaz de converter de forma genérica e independente do domínio o problema de recomendação em um problema de aprendizado de máquina supervisionado.

Capítulo 3 – COFISL (*Collaborative Filtering to Supervised Learning*)

Este capítulo descreverá a solução proposta para os objetivos apontados na seção 1.2: propor uma metodologia para a transformação do problema de filtragem colaborativa em aprendizado de máquina supervisionado e com isso permitindo trata-lo de outra forma. Inicialmente abordaremos a motivação desta transformação e em seguida ilustraremos o problema propondo um workflow para essa transformação. Por fim iremos discutir cada passo desse processo.

3.1 – Introdução

Com a Internet e o barateamento das TICs, surgiram diversos sistemas oferecendo serviços e produtos aos clientes através da web. Diferentemente das lojas físicas, o cliente não possui nenhum contato humano durante a compra, logo o indivíduo não tem a possibilidade de se beneficiar de nenhum instrumento que recomende ou indique um produto. Devido a esse problema, essa busca se transforma impessoal e objetiva. Outra característica que agrava essa situação é a capacidade destas lojas possuírem uma quantidade ilimitada de produtos dentro da loja virtual.

Devido a essa impessoalidade e a possibilidade de possuir uma quantidade muito grande de produtos ou serviços, que dão a impressão do sistema possuir “prateleiras infinitas”, podem gerar dificuldades aos clientes na hora da escolha. Visto esse problema, surgiram os sistemas de recomendação que possuem o objetivo de tornar mais pessoal o consumo na internet, ou seja, tornar mais eficiente à extração e a descoberta de informações dentro de um conjunto enorme de dados.

A filtragem colaborativa consiste na tarefa de recomendar tendo como base somente a interação dos usuários sobre os itens, que é representada através de uma nota. Esse tipo de recomendação independe do conhecimento explícito do conteúdo do item e do próprio usuário. Essa relação entre as duas entidades, que normalmente é representado através de uma matriz, possui características esparsas, ou seja, pequenas quantidades de relações definidas, comparada com as possíveis.

Em comparação com as outras abordagens de recomendação, a filtragem colaborativa é uma das mais utilizadas por sua simplicidade e eficiência. Esse tipo de técnica pode ser aplicada em diversos contextos por causa da sua simplicidade, necessitando somente do

conjunto de preferências dos usuários sobre os itens. Existem diversas classes de algoritmos dentro da filtragem colaborativa, sendo que a baseada em modelo vem se destacando, porque se comparado com os baseados em memória, ela lida melhor com a escalabilidade em grandes conjuntos de dados, além de possuir um melhor desempenho na recomendação.

O aprendizado de máquina supervisionado possui o objetivo de inferir uma função a partir dos dados, sendo que para cada instância dos dados, existe um conjunto de características que é associado um valor que se deseja prever. Então o algoritmo utilizaria um conjunto de exemplos para poder construir uma função a qual generalizaria o problema, e assim teria a possibilidade de prever o valor para uma nova instância. Esse processo é demonstrado na Figura 16.

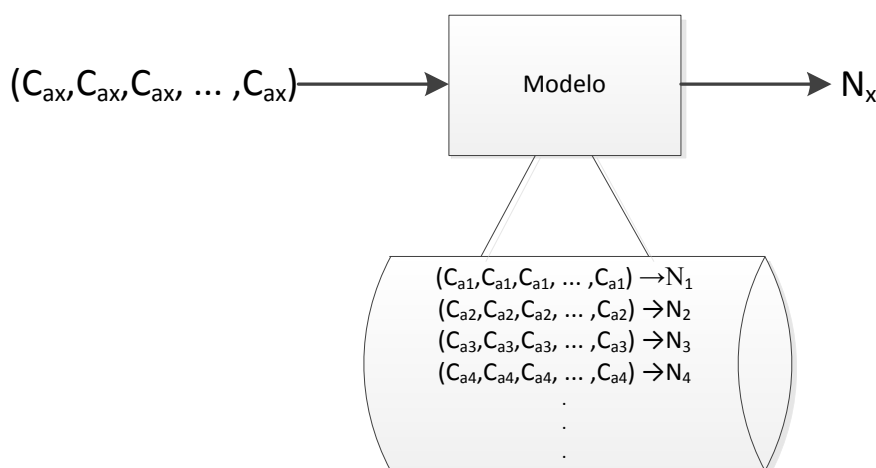


Figura 16 – Aprendizado de máquina supervisionado.

Devido à configuração da filtragem colaborativa, a aplicação de uma técnica de aprendizado de máquina torna-se não trivial. Nesse tipo de técnica, espera-se ter como entrada uma amostra representada por um conjunto de atributos. O problema é definir esse conjunto de atributos para cada relação existente definida por um usuário e por item no contexto da filtragem colaborativa, pois nela o único conhecimento é a nota.

Na abordagem baseada em conteúdo, a aplicação de técnicas de aprendizado de máquina se torna mais fácil, porque a construção de um conjunto de atributos é facilitada devido ao conjunto de informações existente sobre o usuário e o item. Através delas é possível construir o conjunto de treinamento utilizando as informações consideradas importantes para a aplicação.

A filtragem colaborativa baseada em modelo é uma classe de algoritmos que utiliza técnicas de mineração de dados e aprendizado de máquina para construir um modelo matemática que possui o objetivo de encontrar padrões no conjunto de treinamento. Por causa da sua configuração, os algoritmos de aprendizados de máquinas precisam ser adaptados para serem utilizados nesse problema específico e com isso tornando a sua aplicação não trivial.

Como visto nos trabalhos relacionados, existem trabalhos que definem métricas sobre as relações dos usuários e itens para a construção de um conjunto de atributos. Normalmente é utilizado para tal a média das notas dos usuários e do item, a quantidade de itens avaliados pelo usuário e outras medidas. O problema dessa solução é a dificuldade em selecionar as melhores técnicas que melhor caracterizem os dados.

O grande desafio é definir um conjunto de transformações na matriz de preferências dos usuários e com isso ser capaz de construir um novo espaço onde as dimensões estão definidas pelas características inerentes dos usuários e dos itens. Através desse novo espaço seria possível aplicar diretamente os métodos de aprendizado supervisionado. No entanto, a transformação não pode depender de alguma informação sobre o domínio ou de um especialista para ser realizada para assim se tornar genérica em qualquer contexto utilizado por um sistema de recomendação.

3.2 – Definição do Problema

Nesta seção, iremos formalizar o problema abordado na seção anterior. Como discutido, o objetivo é criar uma metodologia que seria capaz de converter o problema definido pela filtragem colaborativa para um problema clássico de filtragem colaborativa, ou seja, definir uma transformação que seria capaz de construir um conjunto de treinamento sobre as informações das relações dos usuários e dos itens.

No contexto da filtragem colaborativa, existe um conjunto de usuários U e um conjunto de itens V . Cada usuário U_i pode definir para um subconjunto de itens V' um valor r , sendo que este definido dentro do conjunto de preferência R , que expressa o seu interesse sobre cada item V_j pertencendo a esse subconjunto, ou seja, a relação seria definida como $\langle U_i; V_j \rangle \rightarrow r$.

O objetivo é conseguir definir uma metodologia que possa definir um conjunto de transformações, com o objetivo de construir um vetor de características \overline{W}_{ij} de forma genérica, e possa mapear todas as relações definidas por um usuário e um item. Desta forma, seria possível aplicar diretamente qualquer algoritmo de aprendizado de máquina supervisionado. Essa conversão é exemplificada em Tabela 4.

Tabela 4 - Conversão da filtragem colaborativa para aprendizado supervisionado.

$\langle U_i; V_j \rangle \rightarrow r$	$\overline{W}_{ij} \rightarrow r$
$\langle 1; 3 \rangle \rightarrow 1$	$(W_1, W_2, W_3, \dots) \rightarrow r$
$\langle 3; 4 \rangle \rightarrow 2$	$(W_6, W_7, W_8, \dots) \rightarrow r$
$\langle 6; 7 \rangle \rightarrow 3$	$(W_{11}, W_{12}, W_{13}, \dots) \rightarrow r$
$\langle 7; 7 \rangle \rightarrow 2$	$(W_{16}, W_{17}, W_{18}, \dots) \rightarrow r$
$\langle 1; 6 \rangle \rightarrow 5$	$(W_{21}, W_{22}, W_{23}, \dots) \rightarrow r$

3.3 – Proposta

O presente trabalho propõe uma metodologia para transformar o problema da filtragem colaborativa em um problema de aprendizado supervisionado (COFILS – *Collaborative Filtering to Supervised Learning*). Essa transformação consiste em um conjunto de operações com intuito de construir um novo espaço que é definido pelas características dos usuários e dos itens, utilizando a matriz de preferências. A previsão de uma nota, utilizando o novo espaço definido pelas operações, se torna um problema de reconhecimento de padrão, sendo assim possível, a aplicação das técnicas de aprendizado de máquina. Esse processo é ilustrado na Figura 17.

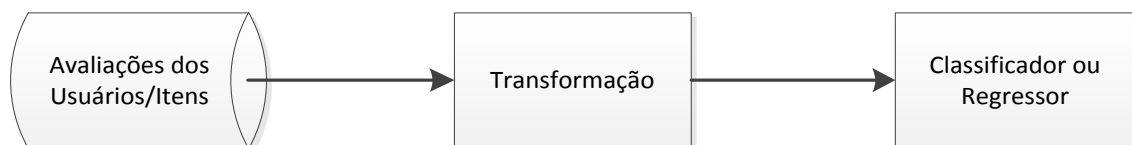


Figura 17 – Processo de transformação da matriz de notas para um classificador ou regressor.

Um caminho comum para lidar com a matriz esparsa é a utilização de redução de dimensionalidade. Essa técnica usualmente transforma uma matriz em outras através de

decomposições. Existem diversos métodos da área da álgebra linear com objetivo de decompor uma matriz, como exemplo o SVD, PCA, Tensores, QR e outros (KOREN *et al.*, 2009). Essas técnicas normalmente são utilizadas para a resolução de sistemas lineares e atualmente vem sendo amplamente estudada a utilização delas para identificar as variáveis latentes dos dados.

A área que estuda as variáveis latentes sobre os dados é chamada de *Latent Semantic Analysis* (LSA). Nela existem vários métodos utilizados no processamento de linguagem natural, e possuem o objetivo de analisar as relações entre os documentos e os seus termos (DEERWESTER *et al.*, 1990). Abordagens similares foram propostas em sistemas de recomendações para melhorar a precisão da previsão. Em (SARWAR *et al.*, 2000), um algoritmo baseado na Decomposição por Valores Singulares foi utilizado para realizar a regressão dos usuários-itens não avaliados.

As variáveis latentes, de alguma forma, tentam expressar as relações entre os dados. Utilizando o contexto de recomendação de filmes como exemplo, é aplicada a decomposição na matriz de notas dadas dos usuários aos filmes a fim de produzir duas matrizes (U e V), onde U é a matriz dos fatores do usuário e V é a matriz dos fatores dos filmes. Com isso os usuários e os filmes são definidos como vetores de fatores. Cada fator poderia ser correlacionado como a categoria de um filme, como comédia, horror, drama e outros. (KOREN *et al.*, 2009)

Através desse conjunto de técnicas é possível representar os usuários e os itens em um definido por seus fatores. Contudo, mesmo utilizando essa representação para cada um, não seria possível construir um conjunto de treinamento diretamente com intuito de ser utilizado para treinar um modelo através de alguma das técnicas de aprendizado supervisionado, porque não existe uma correlação entre os espaços.

Com os fatores dos usuários e dos itens falta definir um modelo que correlacione essas características com as notas com intuito de construir os dados para o treinamento de um classificador ou regressor. Esse modelo definiria como esses fatores se relacionam, e este definiria uma combinação que teria como finalidade a construção de um espaço único onde esses fatores definiriam as notas.

Assim podemos definir que a transformação ilustrada na Figura 17, pode ser dividida em duas etapas. A primeira seria a extração das variáveis latentes dos usuários e dos itens. Já a segunda seria a construção de um modelo que utilizaria as informações extraídas da primeira etapa e as correlacionaria com as notas, e assim seríamos capaz de criar um conjunto de treinamento para poder ser aplicado em um algoritmo de aprendizado de máquina. Na Figura 18 mostra a extensão do processo de transformação estendendo ele com esses dois novos passos.

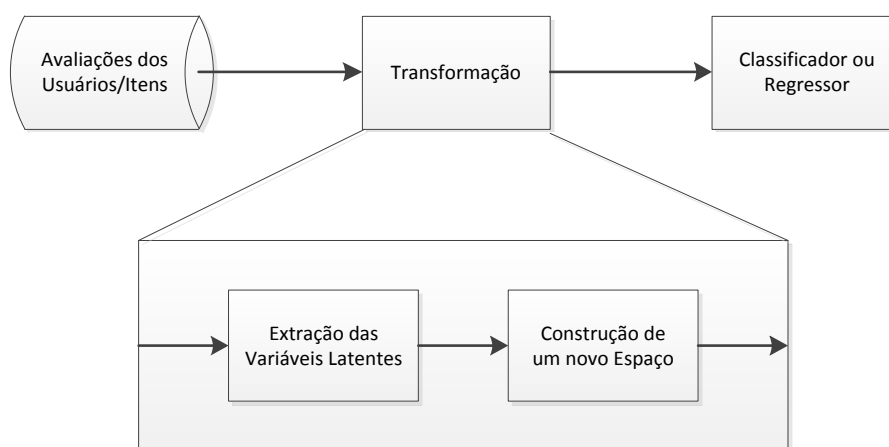


Figura 18 – Proposta

Na subseção a seguir, serão discutidas as duas etapas da proposta, sendo que na primeira será discutido o problema da extração das variáveis latentes. Nessa etapa, serão abordados os problemas envolvidos na extração dessas características devido à falta de dados e às características do problema. Por último será discutido a construção de um modelo que utilizaria essas informações para a realização da predição de uma nota utilizando um algoritmo de aprendizado de máquina supervisionado.

3.3.1 – Transformação

3.3.1.1 - Extração das Variáveis Latentes

Um sistema de recomendação, especificadamente o contexto da filtragem colaborativa, contém duas entidades: usuários e itens. Essas duas possuem uma interação explícita, sendo que essa relação é dada por um valor dentro de um elemento do conjunto de preferência. Normalmente, seu significado é uma nota de um usuário para um item. O

objetivo dos sistemas de recomendação é recomendar um item na qual o usuário gostará e isto os algoritmos utilizam a previsão das relações não definidas para realização desta tarefa.

No modelo definido pela filtragem colaborativa, as notas dos usuários para os itens, ou seja, as interações explícitas são as variáveis observadas do problema. Considerando que esse problema pode ser representado por um grafo bipartido não direcionado (MELLO *et al.*, 2010), exemplificado na Figura 19, as únicas informações capturadas do sistema é o conhecimento de algumas arestas $\langle U_i; V_j \rangle$ e o seu respectivos pesos.

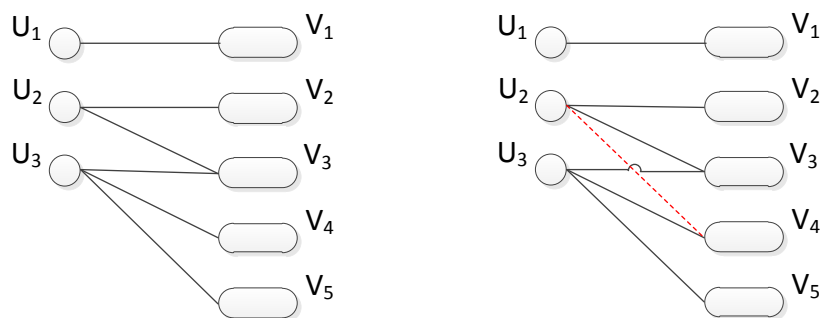


Figura 19 - Representação do modelo de recomendação através de um grafo bipartido não direcionado.

Logo, pode-se definir que para cada usuário u e item v pode existir uma relação observada representada pela aresta $\langle U_i; V_j \rangle$ ou uma relação que gostaríamos de prever. Então podemos representar essa relação para ser aplicado um algoritmo de aprendizado de máquina como $U_i \times V_j \rightarrow \langle U_i; V_j \rangle$. O grande desafio é que o lado esquerdo não está definido explicitamente, ou seja, não existem informações observadas sobre os usuários e sobre os itens, com isso impedindo a aplicação direta dos algoritmos de aprendizado supervisionado.

Entretanto pode-se considerar que tanto o usuário quanto o item possuem certas características que podem representar o seu comportamento, mas elas não podem ser observadas diretamente. Esse tipo de variável é chamado de variável latente e são aquelas que não são observadas diretamente, mas podem ser inferidas através de um modelo matemático ou a partir de outras variáveis observadas, que nesse caso seria a preferência do usuário pelo item.

Em (DEERWESTER *et al.*, 1990), propõe-se uma teoria de representação do espaço através de variáveis latentes chamado Análise Semântica Latente (LSA). Ela possui como

origem dentro da área da Busca e Recuperação de Informação, e possui o objetivo de analisar as relações entre documentos e palavras para produzir um conjunto de fatores relacionados a cada uma das duas entidades. LSA é baseada na utilização de decomposição em valores singulares (SVD) que é uma técnica de decomposição de matriz.

O problema de analisar as relações entre documentos e palavras através da sua frequência pode ser comparado ao problema da filtragem colaborativa. O objetivo do LSA é utilizar a matriz de notas para encontrar as variáveis ocultas do usuário e do item e assim ser possível representar o problema em um espaço comum de características.

A grande diferença dentro dessas duas áreas é que na filtragem colaborativa a matriz de preferência de usuários para itens não está definida totalmente. Outro complicador é o fato da quantidade das notas possuir um valor muito baixo tendo como comparação a quantidade total de relações possíveis e assim dificultando a aplicação dos algoritmos de decomposição de matrizes. O grande desafio para a extração das variáveis latentes é representar o conjunto de preferência e para isso definir as relações desconhecidas.

A primeira abordagem, sendo a mais ingênua, é considerar as relações desconhecidas como um valor distinto do conjunto de preferências. Um exemplo utilizando o conjunto de preferência com os valores inteiros entre um e cinco seria considerar essa relação como zero, como mostra na Figura 20. A desvantagem dessa solução é a inserção de uma informação que destoa dentro da matriz. Esses estariam com um valor abaixo em comparação com o menor valor do conjunto de preferência e tendo como consequência a mudança da semântica das notas, ou seja, o item desconhecido seria um que o usuário não tenha gostado. Com isso acrescentaria um grande erro na extração das variáveis latentes.

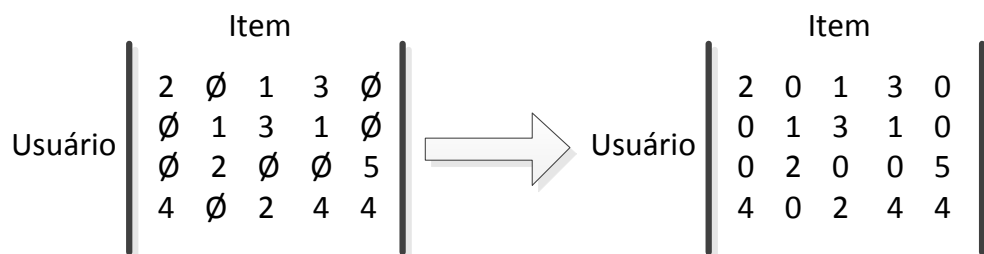


Figura 20 - Exemplo de substituição dos valores desconhecidos por zero.

Outra abordagem e similar a primeira é considerar os valores desconhecidos como a média global de notas. Essa solução considerada que o valor desconhecido tende a média em

sua maioria. Outra vertente dessa abordagem é utilizar a média do usuário ou do item como substituto do valor desconhecido, e com isso minimizando o erro global da extração das variáveis. Nesse último caso a maior dificuldade é quando existir um usuário e/ou um item que possua nenhuma avaliação ou possua uma quantidade muito pequena, principalmente quando exista um desvio padrão alto para este. Para esses casos poderia ser utilizada a média global como substituto.

Em todos os casos, estamos tentando encontrar uma representação dos valores desconhecidos que minimize o erro na extração das variáveis latentes. A melhor representação da matriz seria prever cada valor desconhecido, ou seja, estamos completando os valores. Em, (CANDÈS & RECHT, 2009) demonstra-se que a matriz dos dados da filtragem colaborativa pode ser aproximada por uma matriz com o posto baixo. Então, é possível aplicar as técnicas de completar matriz para representar os valores desconhecidos.

Por fim, qualquer algoritmo de filtragem colaborativa poderia ser utilizado para a finalidade discutida. O grande problema dessa abordagem seria o aumento da complexidade da solução, podendo não ser útil à aplicação do *framework* caso o ganho do resultado não faça jus a este aumento, principalmente se o algoritmo de filtragem colaborativa possuir um grande custo para prever somente uma nota, porque para ser aplicado existiria a necessidade de prever todas as notas desconhecidas.

Como exemplo de extração das variáveis latentes, a Figura 21 ilustra a decomposição utilizando a técnica Decomposição por Valores Singulares sobre a matriz de notas R . Para essa operação iremos considerar dois fatores e após a sua aplicação será produzido a matriz de fatores do usuário (U) e do item (V). As duas colunas das duas matrizes representam os dois fatores e para cada linha é representado um usuário e um item respectivamente.

falta definir um modelo que irá correlacioná-las com suas respectivas notas e por fim será aplicado um algoritmo de aprendizado supervisionado para realizar a previsão.

Na filtragem colaborativa, pode-se definir que para cada usuário u e item v pode existir uma relação observada representada pela aresta $\langle U_i; V_j \rangle$ ou uma relação que gostaríamos de prever. Então podemos representar essa relação para ser aplicado em um algoritmo de aprendizado de máquina como $U_i \times V_j \rightarrow \langle U_i; V_j \rangle$. O grande desafio é que o lado esquerdo não está definido explicitamente, ou seja, não existem características observadas pelo problema que poderiam exemplificar a nota através do seu usuário e do seu item.

No passo anterior, foi visto que existem características não observadas diretamente, *i.e.* variáveis latentes, que definem o comportamento do usuário e do item, e existem diversos métodos que extraem essas variáveis da matriz de preferência. Então, podemos definir que após aplicar as técnicas de extração de variáveis latentes, discutida anteriormente, teremos para cada usuário i um vetor de características \bar{C}_{U_i} e para cada usuário j o vetor \bar{C}_{V_j} .

Com esses dois vetores a relação $U_i \times V_j \rightarrow \langle U_i; V_j \rangle$ fica definida do lado esquerdo, ficando com $\langle \bar{C}_{U_i} | \bar{C}_{V_j} \rangle \rightarrow \langle U_i; V_j \rangle$, ou seja, existe um conjunto de características de entrada definidas pelas características do usuário e do item $\langle \bar{C}_{U_i} | \bar{C}_{V_j} \rangle$ que implica em uma nota $\langle U_i; V_j \rangle$. Através dessa construção desse novo espaço será possível construir um conjunto de treinamento para ser aplicado em um algoritmo de aprendizado supervisionado.

Como visto anteriormente, os algoritmos de aprendizado de máquina supervisionados são conhecidos por serem capazes de realizar mapeamentos complexos entre a entrada e a saída, ou seja, tais algoritmos são capazes de encontrar uma função Θ que será responsável por mapear um vetor de característica \bar{C} em outro vetor de saída \bar{S} . Dependendo do algoritmo de aprendizado supervisionado utilizado, essa função se torna linear ou não.

Com a aplicação de um algoritmo de aprendizado supervisionado no novo espaço definido como $\langle \bar{C}_{U_i} | \bar{C}_{V_j} \rangle \rightarrow \langle U_i; V_j \rangle$, o problema se torna a construção de um modelo que irá encontrar a melhor função de mapeamento Θ . Então, podemos rescrever o problema como $\Theta(\bar{C}_{U_i}, \bar{C}_{V_j}) = \langle U_i; V_j \rangle$.

Na Figura 23 exemplifica-se a proposta de transformação do problema da filtragem colaborativa em um conjunto de treinamento para ser aplicado em um algoritmo de

aprendizado supervisionado. No primeiro passo, a média global é utilizada para definir os valores desconhecidos. Após, é realizada a aplicação da decomposição por valores singulares para extrair as variáveis latentes dos usuários e dos itens. Por fim, é construído um conjunto de treinamento utilizando as variáveis latentes e a matriz de preferência para ser aplicado em um classificador/regressor.

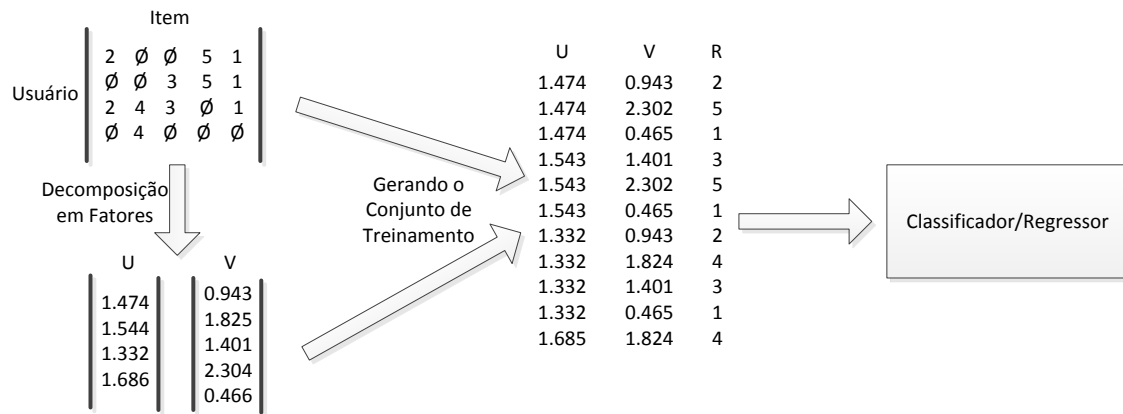


Figura 23 – Exemplo da proposta de transformação.

Para exemplificar desenhamos o gráfico $U \times V$ na Figura 24. Como na extração das variáveis latentes foi utilizada somente uma característica para U e uma para V , o mapeamento foi direto, sendo o eixo horizontal a variável latente do usuário e o eixo vertical a do item. Como se pode ver, este espaço possui um padrão que é separável e, portanto, um algoritmo de aprendizado supervisionado é capaz de separar as notas neste espaço.

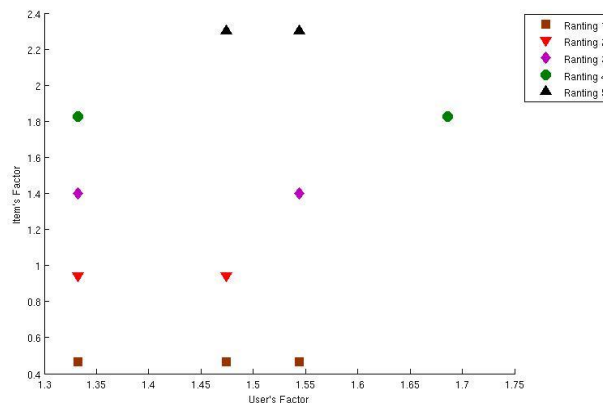


Figura 24 – Representação visual do espaço do conjunto de treinamento gerado no exemplo anterior.

A variação dos parâmetros, como exemplo, o número de fatores para cada variável latente, a técnica utilizada para a extração dessas variáveis ou a definição dos valores

desconhecidos da transformação proposta podem gerar diversos espaços diferentes. Assim, um classificador/regressor pode variar o seu desempenho dependendo destes parâmetros, uma vez que a escolha deles afeta, como por exemplo, a distribuição do espaço de características.

Esse problema do classificador depender dos parâmetros utilizados na transformação é demonstrado na Figura 25. Nela foi utilizado o exemplo anterior, mas utilizando a média do usuário ao invés da média global. Note que se gerou outra distribuição das instâncias e dependendo dessa distribuição pode-se beneficiar alguns classificadores/regressores do que outros.

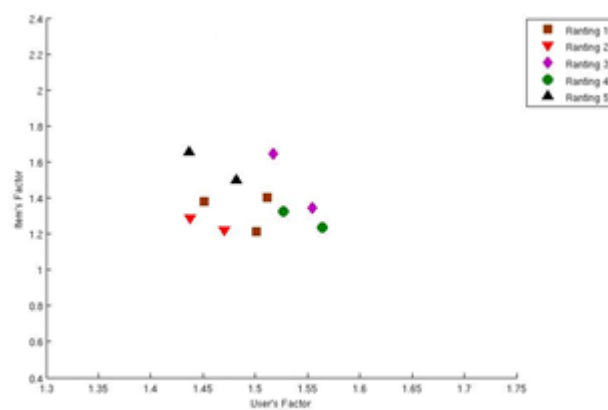


Figura 25 - Representação visual do espaço do conjunto de treinamento gerado no exemplo anterior utilizando a média do usuário.

Capítulo 4 – Avaliação Experimental

Neste capítulo, primeiramente, serão apresentados os objetivos dos experimentos, e em seguida a metodologia utilizada. Será detalhada a organização dos experimentos, como foram executados e o seu resultado. Por fim, para cada um dos experimentos será discutido os resultados e será realizada uma análise geral sobre os mesmos.

4.1 - Objetivos dos Experimentos

Na seção anterior foi apresentada a proposta do presente trabalho de uma transformação com objetivo de converter o problema da filtragem colaborativa para ser possível aplicar os algoritmos clássicos de aprendizado de máquina supervisionado. Para tal, foram propostos passos para contemplar a transformação e o primeiro envolve a extração das variáveis latentes. Por último, construir um novo espaço envolvendo as características latentes dos usuários e dos itens. Nele será montado o conjunto de treinamento com intuito de ser aplicado um algoritmo de aprendizado supervisionado.

Existem diversas variações tanto na primeira etapa quanto na segunda etapa devido às inúmeras técnicas que poderiam ser utilizadas para atacar o problema da extração das variáveis latentes, definição dos valores desconhecidos e algoritmos de aprendizado supervisionados. Além dessas, existem diversas configurações e arquiteturas possíveis para cada classificador/regressor e dependendo do arranjo da primeira etapa pode influenciar o algoritmo escolhido. Os objetivos dos experimentos é avaliar cada uma delas e verificar o seu desempenho e estabilidade em diversas bases de dados com características distintas. Por fim, comparar com os métodos clássicos da literatura de filtragem colaborativa.

4.2 - Base de Dados

O conjunto de dados utilizado nos experimentos foi o *MovieLens* (MILLER *et al.*, 2003). Essa base de dados vem de um sistema de recomendação de filmes desenvolvido pelo *GroupLens*⁹ e possui 100.000 avaliações. Nela existem 943 usuários e 1682 filmes, e o seu conjunto de preferência é um número inteiro que varia entre um a cinco. Além das avaliações, existem informações sobre usuários, como a idade e o sexo, e sobre os filmes, tais como título do filme e a data de lançamento do filme. Essas informações não foram utilizadas em nenhum

⁹ <http://www.grouplens.org/>

momento, pois se fosse utilizada alguma informação dos usuários ou itens descaracterizará a proposta como algoritmo de filtragem colaborativa.

Como toda base de sistema de recomendação, ela possui poucas avaliações comparando com todas as possibilidades. A proporção de avaliações com relação ao total, ou seja, o índice de esparsidade é de 6.30%. A distribuição das notas possui o comportamento de uma normal com média 3.5299 e desvio padrão de 1.1257, como mostra na Figura 26.

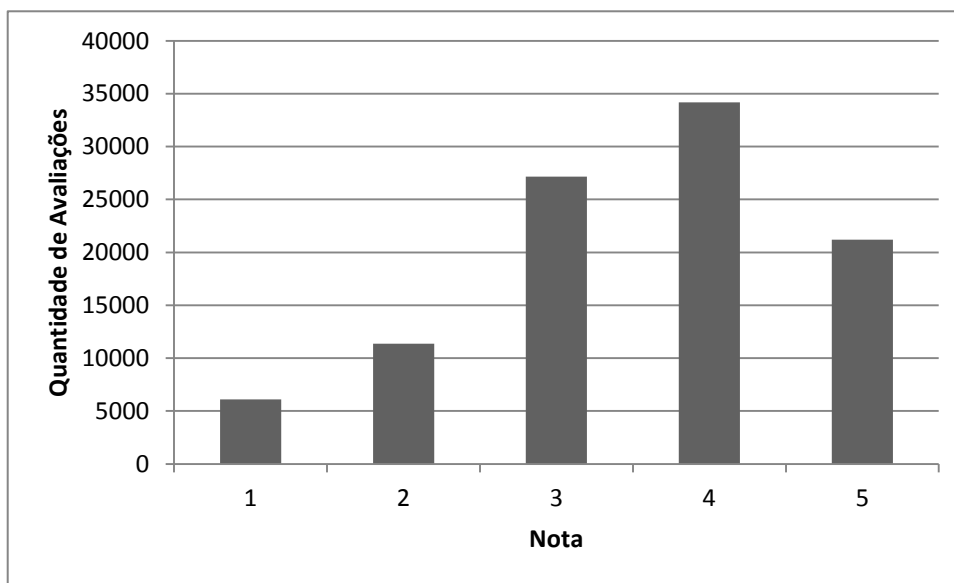


Figura 26 - Quantidade de Avaliações com relação a nota da base *MovieLens*.

A quantidade de avaliações feitas por um usuário possui média de 106.04 avaliações e se comporta como uma distribuição exponencial, como se observa na Figura 27, sendo que a avaliação mínima é de 20 avaliações e a máxima de 737. Essa característica de possuir um valor alto para o mínimo de avaliações por usuário é por causa das características do sistema *MovieLens*. Durante o ato de criação da conta, o usuário é obrigado avaliar essa quantidade mínima.

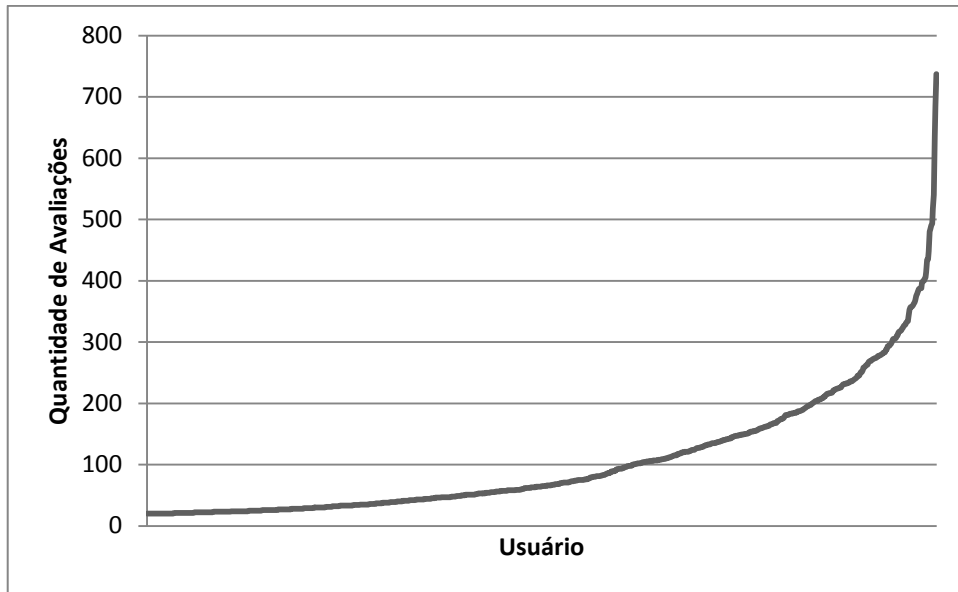


Figura 27 - Quantidade de Avaliações com relação ao usuário da base *MovieLens*.

Já no caso dos filmes a média de quantidade de avaliações por filme é de 59.45 e como no caso dos usuários também possui uma distribuição exponencial como se observa na Figura 28. Diferentemente do caso dos usuários, não existe nenhuma restrição na quantidade mínima de avaliações e nessa base a menor quantidade é de uma avaliação e a máxima é de 583.

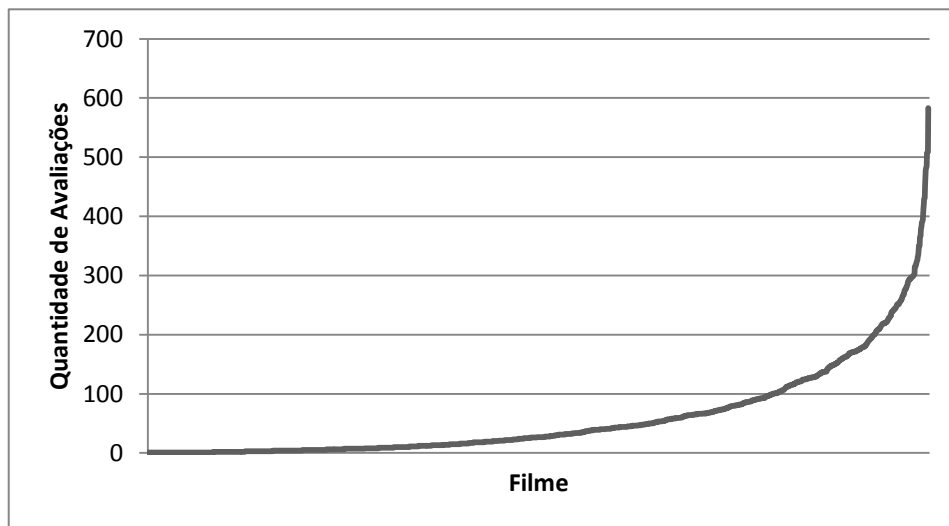


Figura 28 - Quantidade de Avaliações com relação ao filme da base *MovieLens*.

4.3 - Metodologia e Organização dos Experimentos

A metodologia da avaliação experimental da proposta será composta de diversos experimentos com objetivo de avaliar o desempenho da técnica utilizando diversas combinações dos parâmetros. Como o método utilizado para extração das variáveis latentes, o algoritmo de aprendizado supervisionado, número de variáveis e outros, a fim de comparar com as técnicas clássicas da literatura de sistema de recomendação como vizinhos mais próximos, *SVD by Funk* e o *SVD Regularized*.

A metodologia proposta (*COFILS*) possui três etapas: o pré-processamento, a extração das variáveis latentes e a regressão/classificação. A primeira etapa será responsável pela representação dos dados e como corrigi-la para que consiga representar de melhor forma o problema devido ao seu problema com os valores desconhecidos. Nessa etapa foram escolhidas três abordagens, sendo que a primeira seria utilizar os valores desconhecidos como zero e sem normalização. As outras duas seriam normalizar a matriz com relação a média do usuário ou do item e assim os valores desconhecidos tornariam a média do fator normalizador, ou seja, o valor zero. Essa transformação é ilustrada na Figura 29.

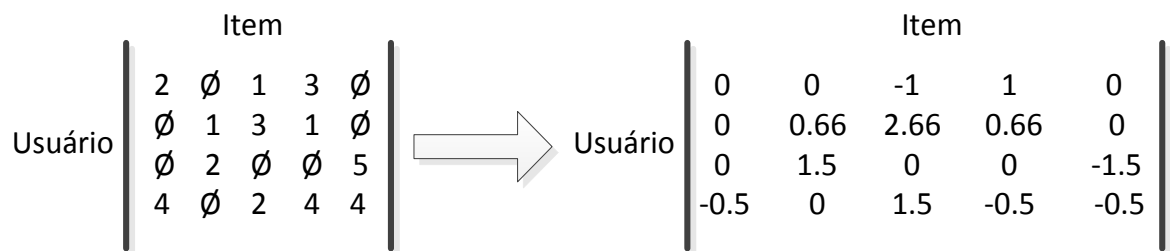


Figura 29 – Exemplo de pré-processamento utilizar a normalização pelo usuário.

Na extração das variáveis latentes será utilizado um conjunto de três técnicas. A primeira é a aplicação da técnica de decomposição de valores singulares (SVD) utilizando a matriz U como variáveis latentes do usuário e a matriz V como do item, e descartando os valores singulares, sendo a mesma estratégia adotada pela área da *Latent Semantic Analysis*.

A segunda é utilizar a técnica de filtragem colaborativa *SVD Regularized*. Essa técnica é um modelo de regressão linear, e nela são definidas duas variáveis (p e q) que correlacionam atributos do usuário e do item respectivamente, como mostra na Equação 16. Por fim é

utilizada a técnica *Partial SVD* da biblioteca *LingPipe*¹⁰. Ela é utilizada em matrizes esparsa, e nela é utilizada a técnica de gradiente descendente para minimizar o erro quadrado desta dos valores conhecidos após a reconstrução da matriz.

Por fim, a última etapa é a aplicação de um algoritmo de aprendizado supervisionado. Para o presente trabalho serão utilizadas três técnicas: aprendizado bayesiano, redes neurais artificiais (ANN) e floresta aleatória. A primeira técnica foi escolhida devido à sua simplicidade, e com isso será possível avaliar a proposta com técnicas simples de aprendizado de máquina. Já a segunda técnica foi escolhida por causa de sua robustez dentro da literatura. A implementação dessas duas técnicas foram utilizadas da ferramenta MATLAB¹¹. Já a floresta aleatória foi escolhida devido à sua precisão e a implementação utilizada foi a *randomforest-matlab*¹².

Resumidamente, existem quatro parâmetros que precisam ser definidos para poder ser aplicado à proposta e são:

- Técnica de pré-processamento e normalização: sem normalização, normalização pela média do usuário e pelo item;
- Técnica de extração de variáveis latentes: *SVD*, *SVD Regularized* e *Partial SVD*.
- Quantidade de variáveis latentes;
- Técnica de aprendizado supervisionado: aprendizado bayesiano, ANN e floresta aleatória.

Além desses quatro parâmetros, existem diversos outros elementos que devem ser definidos para cada técnica de aprendizado de máquina. No caso da rede neural, ela utilizará uma rede de perceptrons com multicamadas, sendo que a arquitetura utilizada é a com duas camadas escondidas e uma saída e a função de transferência é a sigmoide. As configurações do treinamento da rede foram os padrões do MATLAB e podem ser visto na sua

¹⁰ <http://alias-i.com/lingpipe/>

¹¹ <http://www.matlab.com/>

¹² <http://code.google.com/p/randomforest-matlab/>

documentação¹³. Já a floresta aleatória possui um único parâmetro que é a quantidade de árvore de decisões geradas durante o *boosting*.

Dado a grande quantidade de parâmetros que podem influenciar o resultado seriam necessários inúmeros testes. No contexto desse trabalho queremos apenas validar a nossa proposta, de modo que não estamos procurando os parâmetros ótimos – parâmetros que levem a um resultado melhor do que os algoritmos clássicos já é suficiente para mostrar o potencial da proposta.

O primeiro experimento consiste em definir a configuração das técnicas de aprendizado supervisionado que serão utilizadas nos experimentos seguintes, sendo que para a rede neural será para descobrir a quantidade de neurônios na camada escondida e da floresta aleatória a quantidade de árvores de decisões utilizadas no *boosting*. No caso da rede neural, essa variação será da primeira camada de neurônio e a segunda utilizará o piso da metade desse valor. O algoritmo de aprendizado bayesiano não estará presente, porque não precisa de nenhum parâmetro para ser definido.

O problema desse experimento é definir os outros três parâmetros (normalização, técnica de extração de variável latente e quantidade de variáveis latentes) que serão utilizados. Para tal, foi estipulada a utilização da normalização pela média do usuário. A técnica de extração das variáveis latentes que foi utilizada é o SVD. Este foi escolhido devido à sua importância dentro da área da *Latent Semantic Analysis* e simplicidade. O único parâmetro faltante é a quantidade de variáveis latentes.

Com o objetivo de determinar o número mínimo de variáveis latentes que será utilizado foi aplicado o algoritmo SVD na matriz de preferência dos usuários pelos filmes da base *MovieLens* e analisamos o conjunto dos valores singulares S , na Figura 30. Para efeito de visualização calculamos a diferença entre cada valor singular e seu antecessor ($S_x - S_{x-1}$) e pode ser visto na Figura 31.

¹³ <http://www.mathworks.com/products/neural-network/>

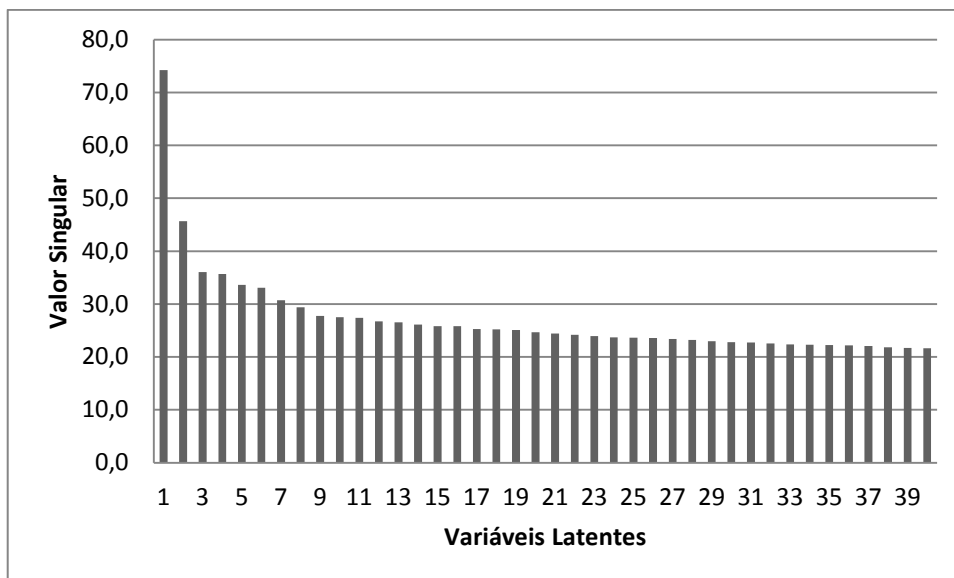


Figura 30 – Quantidade de informação que cada variável latente possui.

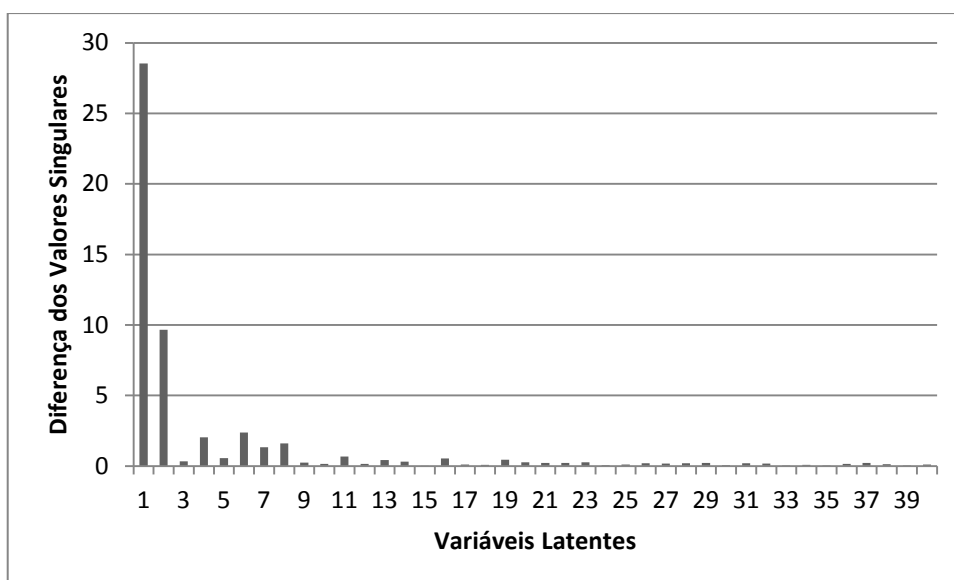


Figura 31 – Ganho de informação que cada variável latente possui.

Observando os valores singulares pode se ver que existe uma grande queda do valor na oitava variável latente e a partir da nona existe pouca variação desse valor. Através dessa análise, foi determinado que fosse fixado esse valor para o experimento tanto para o usuário e quanto para o filme. Assim, determinando o primeiro experimento que terá intuito de definir uma configuração para os algoritmos de aprendizado supervisionados.

O segundo experimento será responsável por avaliar a proposta variando os quatros parâmetros descrito anteriormente. Nele será possível verificar o comportamento da técnica

de aprendizado supervisionado com relação ao número de variáveis latentes, a técnica que extraíram elas e como foi representada a matriz de notas para a extração. Principalmente pelo fato que o desempenho do classificador/regressor estar diretamente correlacionado em função desses parâmetros, uma vez que eles afetam a distribuição dos dados no novo espaço de variáveis latentes.

Após esse experimento, será demonstrado o desempenho de algumas das principais técnicas da filtragem colaborativa com o objetivo de compará-lo com o melhor resultado do experimento anterior. Foram escolhidos quatro algoritmos da filtragem colaborativa tanto utilizando abordagem por memória quanto por modelo que são os vizinhos mais próximos de um usuário e de um item e para cada utilizando a similaridade por cosseno e por Pearson, *SVD by Funk* e o *SVD Regularized*.

Existe uma limitação na técnica dos vizinhos mais próximos, porque existe uma limitação com relação ao número mínimo de vizinhos para realizar a previsão e tendo como consequência a redução da cobertura. Para tal, foi realizado um experimento utilizando o melhor resultado do segundo e verificando o seu comportamento prevendo somente para os usuários ou os filmes que possuem uma quantidade mínima de vizinhos. Com isso verificando se a máquina de aprendizado consegue generalizar bem até para os usuários ou filmes com poucos vizinhos ou se eles influenciam negativamente como acontece na técnica dos vizinhos mais próximos.

Em todos os experimentos utiliza-se a validação cruzada para avaliar a variabilidade de cada algoritmo. Nesse trabalho foi utilizado o *10-fold validation* que consiste em dividir a amostra original aleatoriamente em 10 sub-amostras e para cada uma dessas divisões será aplicado o algoritmo utilizando essas sub-amostras como teste e os demais como treinamento. A vantagem deste método é que todas as observações serão usadas tanto para treino e a validação, quanto para cada observação será utilizada para a validação no mínimo uma única vez.

Resumidamente, serão quatro experimentos com objetivo de avaliar a proposta do presente trabalho:

1. Avaliação das configurações dos algoritmos ANN e floresta aleatória utilizando a normalização pela média dos usuários, oito variáveis latentes e a extração através da decomposição por SVD.
2. Avaliação dos algoritmos aprendizado bayesiano, floresta aleatória e ANN utilizando a configuração do experimento um e variando a normalização, a quantidade de variáveis latentes e a técnica de extração destas.
3. Comparação com as melhores configurações do experimento dois com as quatro técnicas da literatura da filtragem colaborativa: vizinhos mais próximos de um usuário e de um item e para cada utilizando a similaridade por cosseno e por Pearson, *SVD by Funk* e o *SVD Regularized*.
4. Avaliação da melhor configuração do experimento dois variando a quantidade mínima de vizinhos para ser previsto.

4.3 - Avaliação

Para comparar a qualidade dos algoritmos, serão utilizadas duas métricas que são normalmente utilizadas para comparar as técnicas de recomendação, que é o *Mean Absolute Error* (MAE) e *Root Mean Squared Error* (RMSE). A primeira é a métrica mais utilizada (GOLDBERG *et al.*, 2001, HERLOCKER *et al.*, 2004), que calcula a média da diferença absoluta entre as previsões e as reais notas dadas que pode ser vista na Equação 22, onde n é o número total de avaliações de todos os usuários, p_{uv} é a previsão da nota de um usuário u para um item v e r_{uv} é a nota real.

$$MAE = \frac{\sum_{\{u,v\}} |p_{uv} - r_{uv}|}{n}$$

Equação 22 - Mean Absolute Error (MAE).

Já o RMSE é a outra métrica que se tornou popular por causa do *Netflix prize*¹⁴ para avaliação de desempenho de algoritmos de recomendação. A sua maior diferença com relação

¹⁴ <http://www.netflixprize.com>

ao MAE é que ela penaliza os erros grandes em comparado-os aos pequenos. Ela pode ser vista na Equação 23, onde n é o número total de avaliações de todos os usuários, p_{uv} é a previsão da nota de um usuário u para um item v e r_{uv} é a nota real.

$$RMSE = \sqrt{\frac{\sum_{\{u,v\}}(p_{uv} - r_{uv})^2}{n}}$$

Equação 23 – Root Mean Squared Error (RMSE).

4.4 – Resultados

Nesta seção serão apresentados os resultados obtidos em cada experimento. E ao final, será feita uma análise conclusiva destes resultados, avaliando os ganhos utilizando a proposta deste presente trabalho.

4.4.1 – Experimento I

Esse experimento é responsável para avaliar as configurações dos algoritmos ANN e floresta aleatória. Para a extração das variáveis latentes é utilizada a técnica SVD com a matriz de notas normalizadas pela média dos usuários e oito variáveis tanto para o usuário quanto para os filmes, conforme determinado na seção anterior. Segue abaixo os gráficos para cada avaliação utilizando as métricas MAE e RMSE.

1. Rede Neural

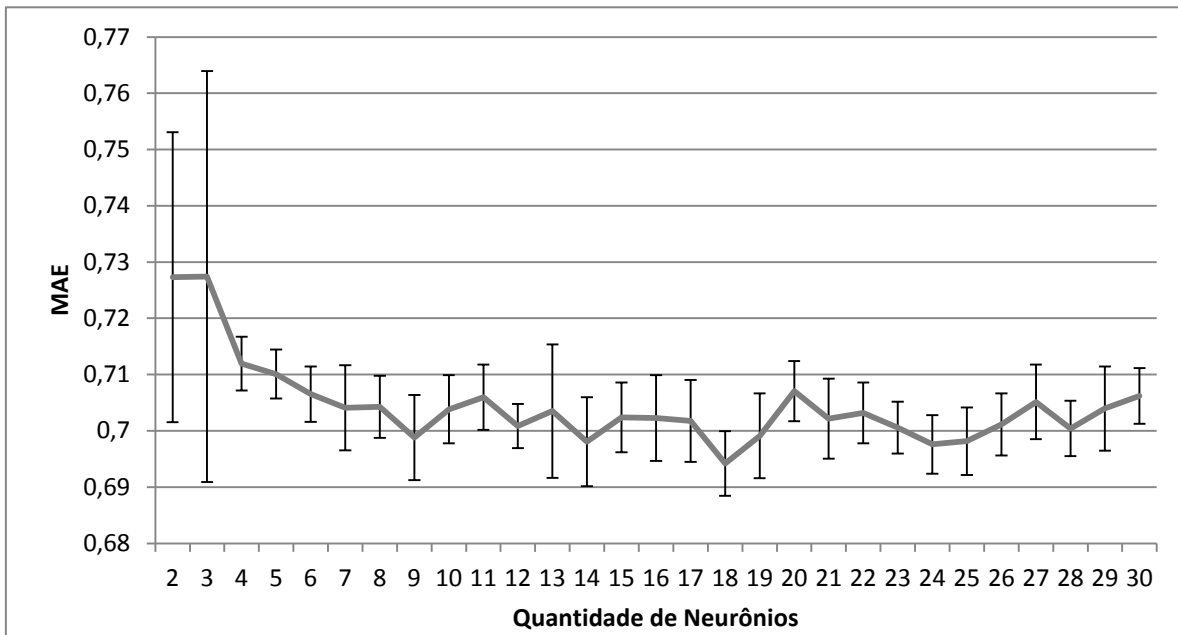


Figura 32 – Avaliação do MAE da técnica ANN variando a quantidade de neurônios.

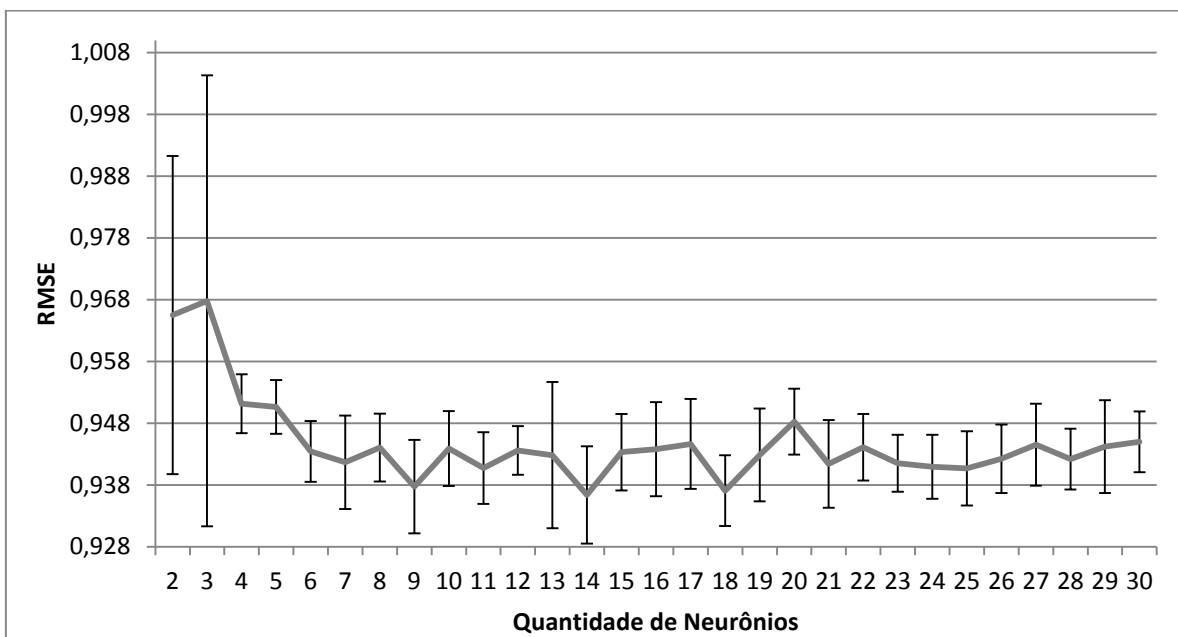


Figura 33 - Avaliação do RMSE da técnica ANN variando a quantidade de neurônios.

2. Floresta Aleatória

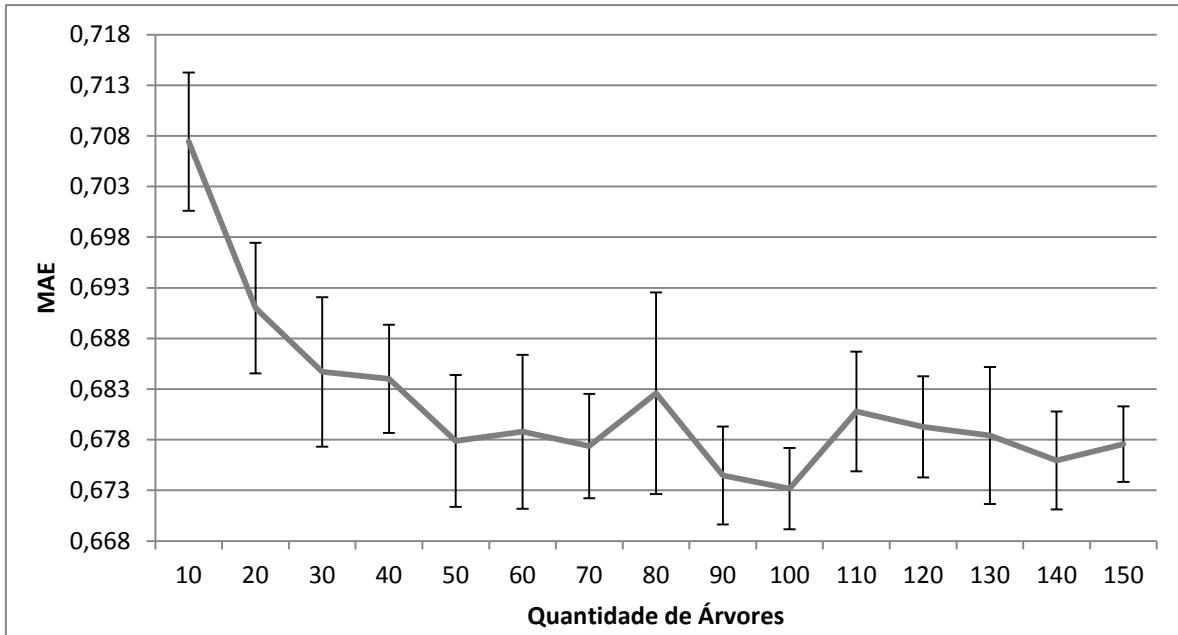


Figura 34 - Avaliação do MAE da técnica floresta aleatória variando a quantidade de árvores de decisões.

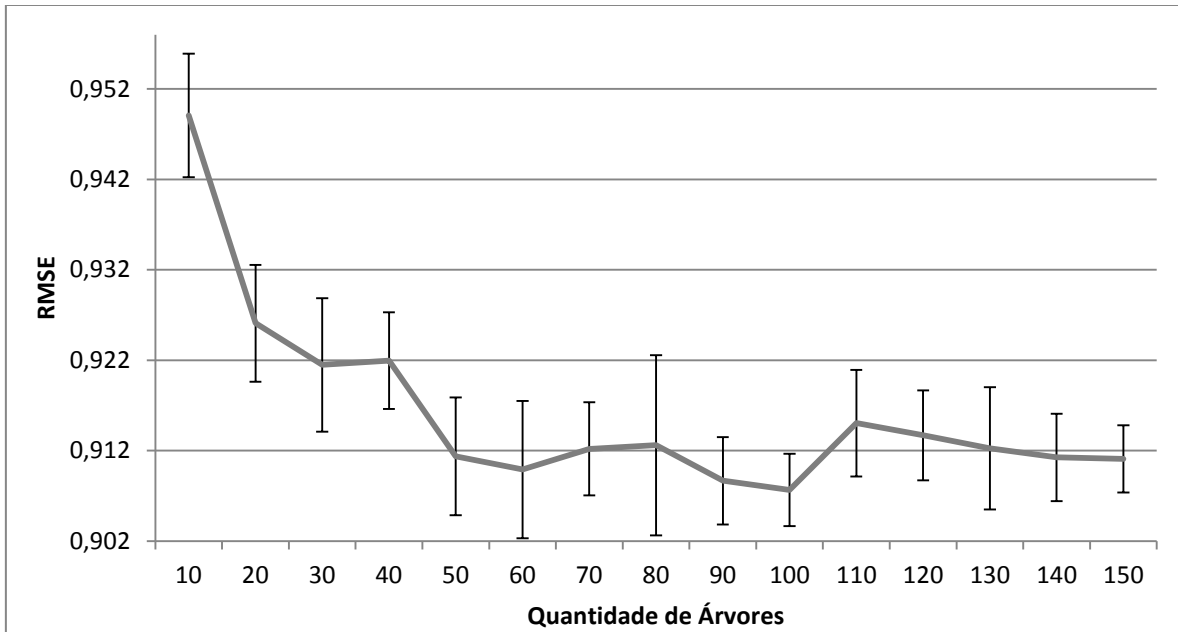


Figura 35 - Avaliação do RMSE da técnica floresta aleatória variando a quantidade de árvores de decisões.

Verifica-se que existe uma tendência de queda do erro (MAE e RMSE) quando aumenta os parâmetros tanto da rede neural quanto da floresta aleatória. No caso da rede neural existe uma estabilização do erro a partir de oito neurônios e na floresta aleatória a

partir de cinquenta árvores. Nos dois casos quando o parâmetro cresce aumenta o tempo de treinamento.

4.4.2 – Experimento II

Este experimento é responsável em avaliar cada algoritmo de aprendizado de máquina variando os parâmetros da proposta como a normalização, técnica de extração das variáveis latentes e a quantidade desta na entrada de cada algoritmo.

O experimento anterior foi responsável de verificar o desempenho da rede neural e floresta aleatória variando as suas configurações. No primeiro verificou-se que a partir de oito neurônios na camada escondida o erro não tende a cair e somente aumentando o tempo do aprendizado. Com isso foi utilizado nesse experimento a configuração de dez neurônios. Na floresta aleatória verificou-se que existe uma forte queda inicial e depois tendendo a estabilizar o erro. Essa estabilização foi a partir do valor de cinquenta florestas de decisões. Com isso esse valor foi mantido no atual experimento.

A seguir segue o experimento dois. Em cada gráfico a representação da legenda será:

- SVD – Decomposição de valor singular;
- SVDM – SVD Regularized;
- SVDL – Partial SVD;
- SN – Sem Normalização;
- CNI – Com normalização pela média do item;
- CNU – Com normalização pela média do usuário;
- NAIIVE - Aprendizado Bayesiano
- ANN – Rede Neural
- FLOREST – Floresta Aleatória

Segue os resultados encontrados.

1. Aprendizado Bayesiano

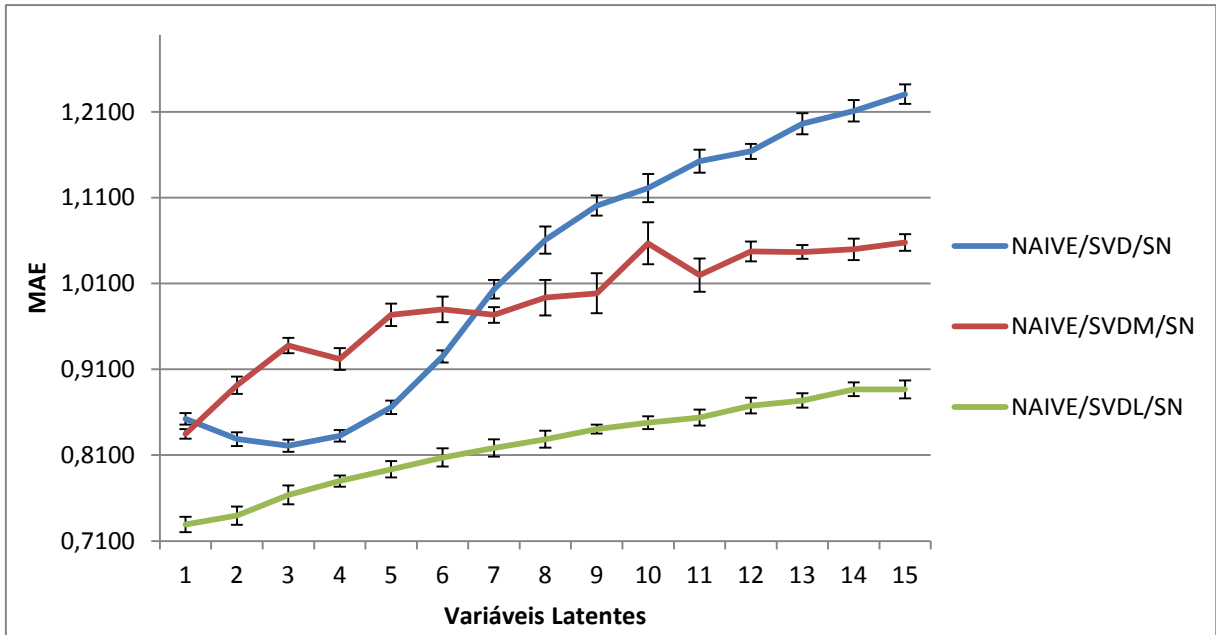


Figura 36 – MAE do experimento utilizando Naive Bayes e sem normalização.

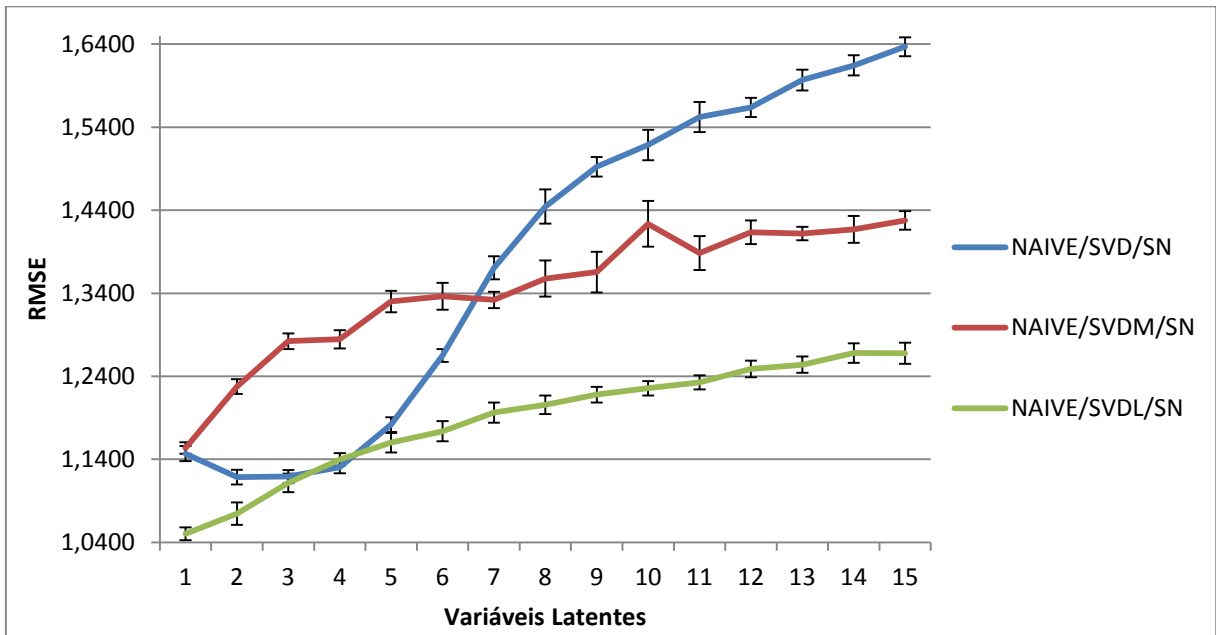


Figura 37 – RMSE do experimento utilizando Naive Bayes e sem normalização.

2. Rede Neural

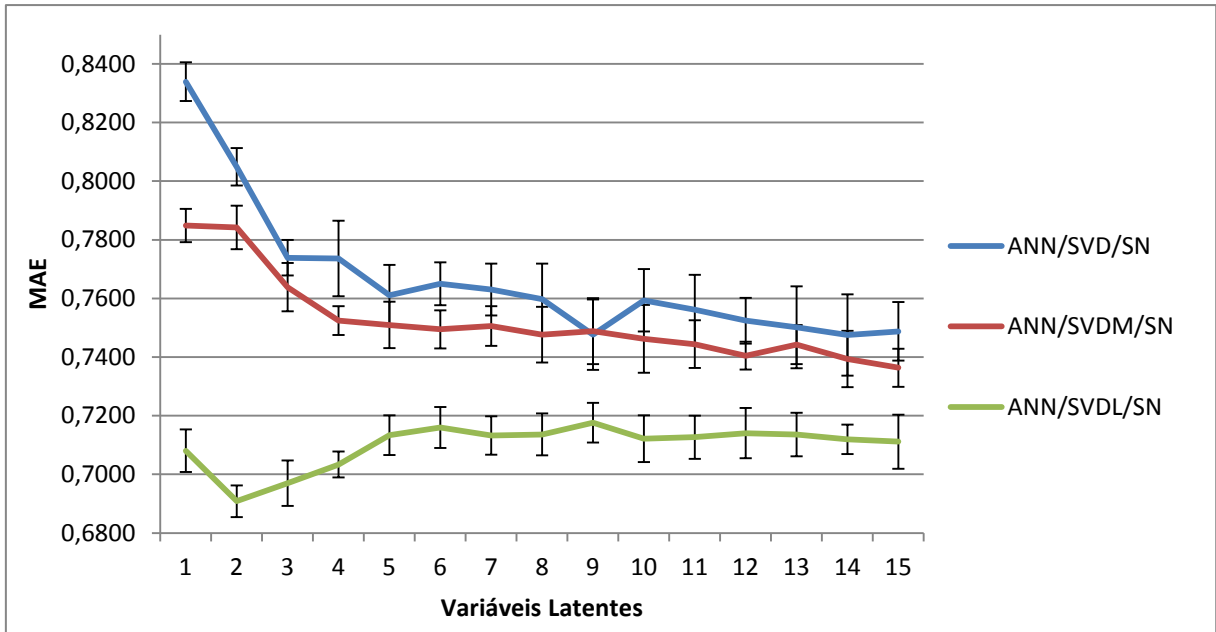


Figura 38 - MAE do experimento utilizando Rede Neural e sem normalização.

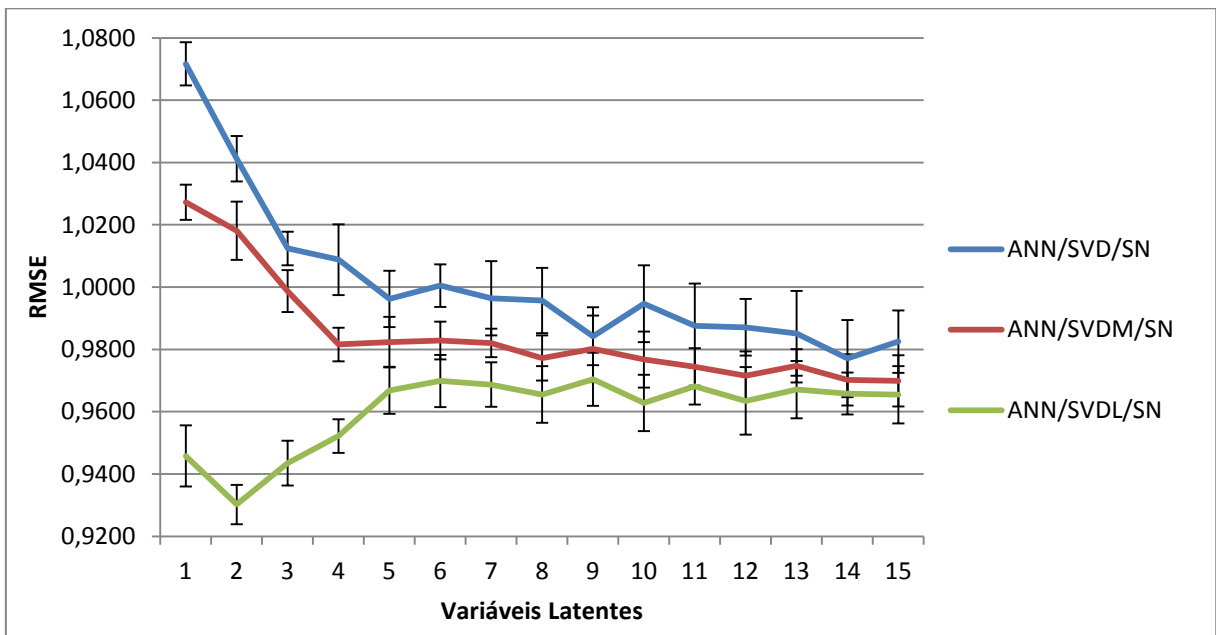


Figura 39 - RMSE do experimento utilizando Rede Neural e sem normalização.

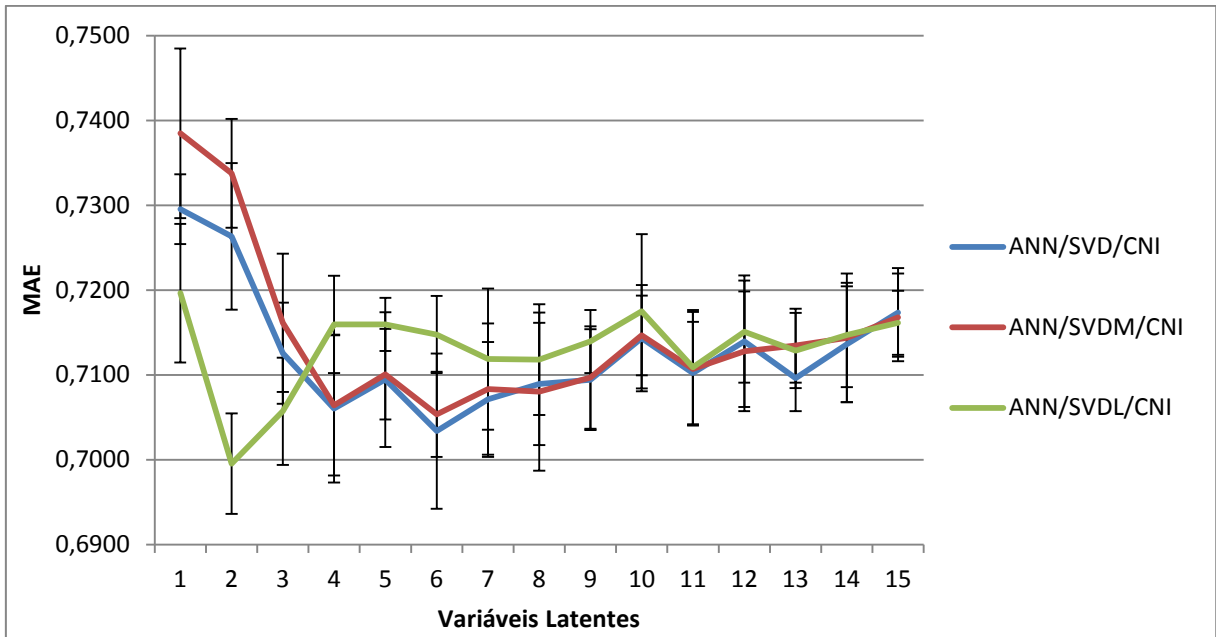


Figura 40 - MAE do experimento utilizando Rede Neural e com normalização através da média do item.

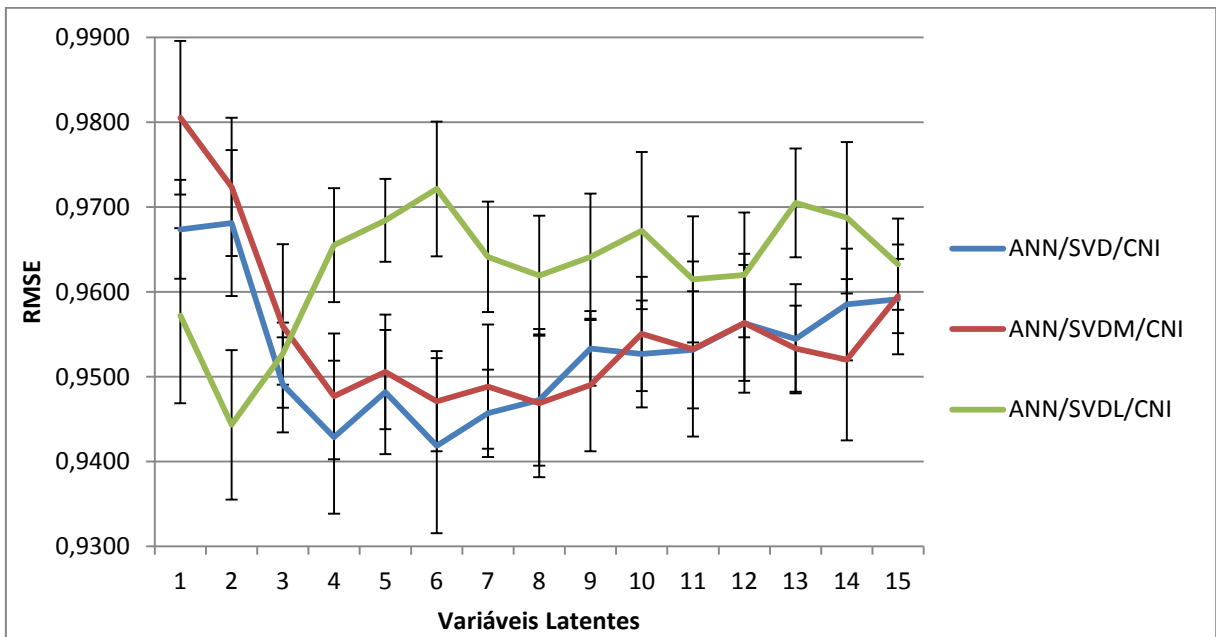


Figura 41 - RMSE do experimento utilizando Rede Neural e com normalização através da média do item.

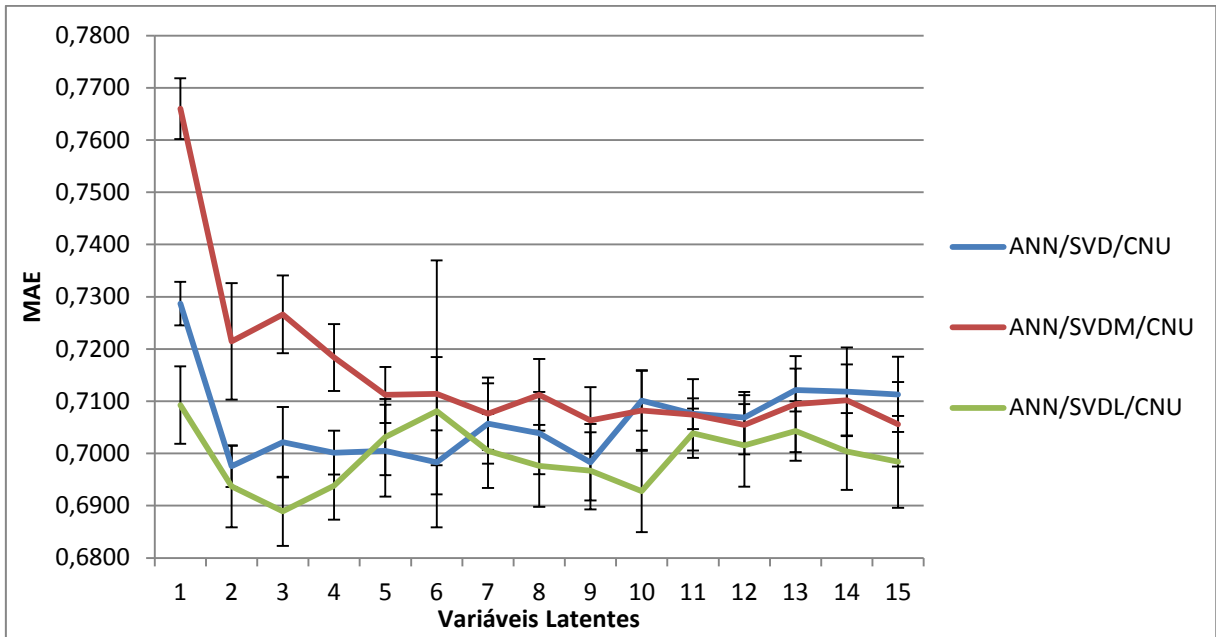


Figura 42 - MAE do experimento utilizando Rede Neural e com normalização através da média do usuário.

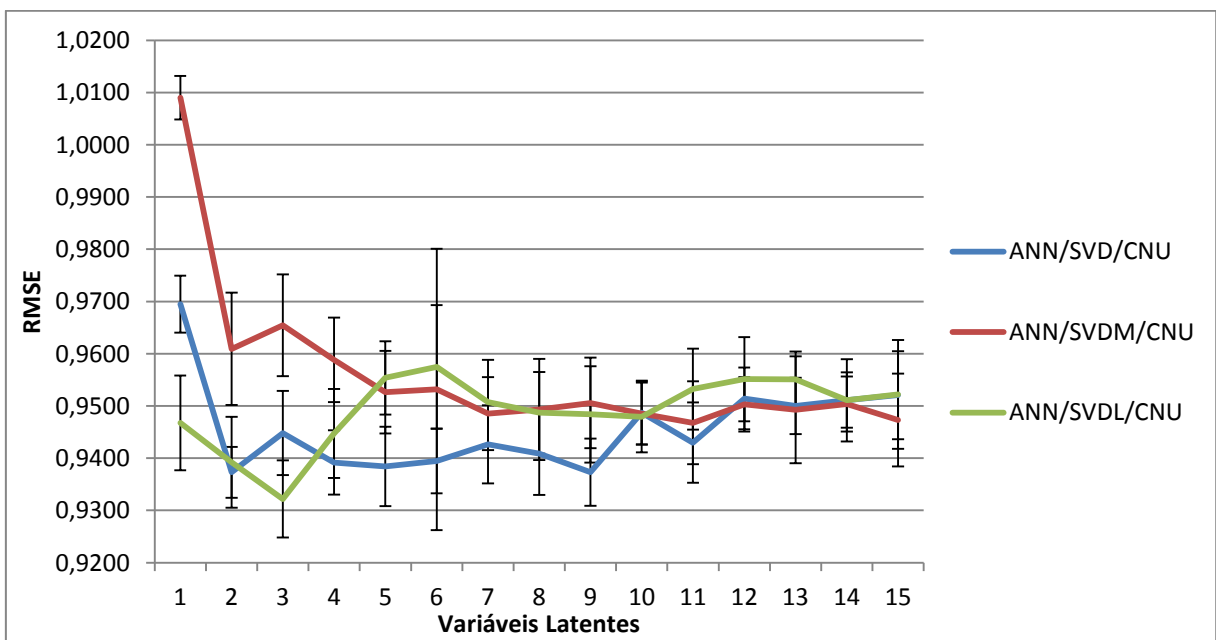


Figura 43 - RMSE do experimento utilizando Rede Neural e com normalização através da média do usuário.

3. Floresta Aleatória

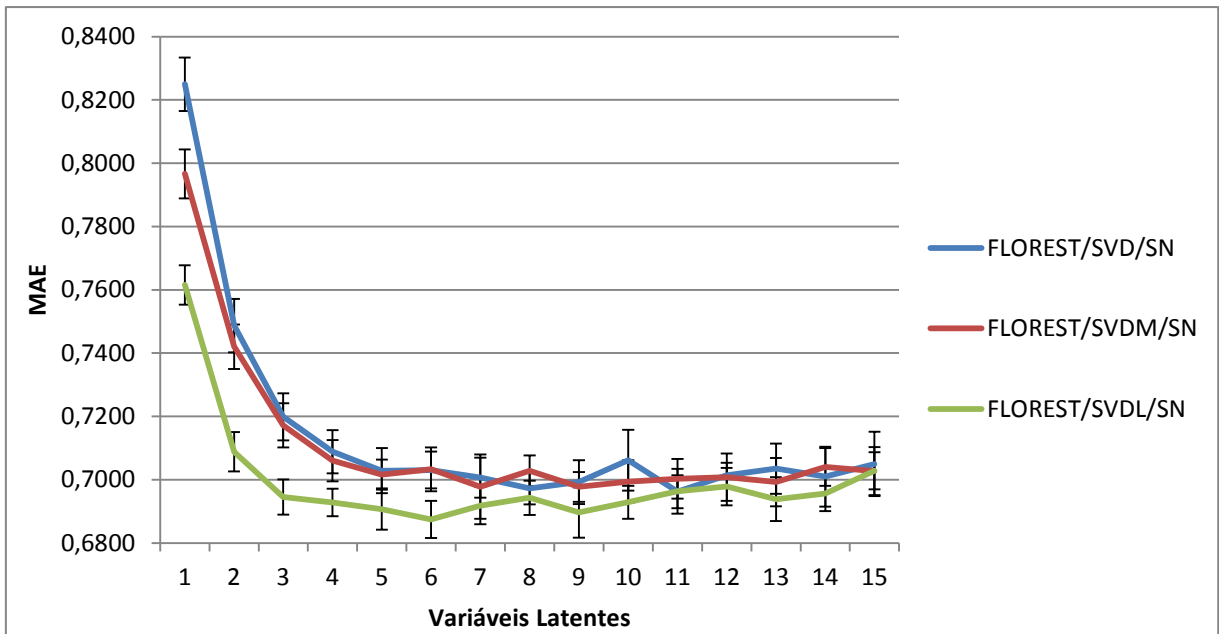


Figura 44 - MAE do experimento utilizando Floresta Aleatória e sem normalização.

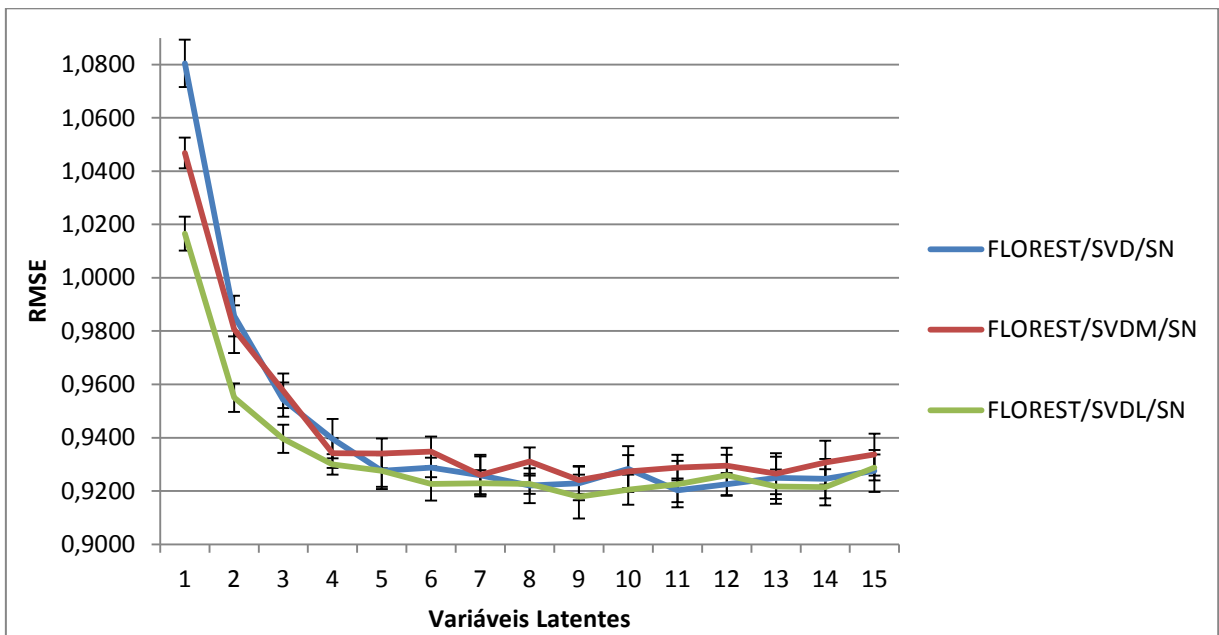


Figura 45 - RMSE do experimento utilizando Floresta Aleatória e sem normalização.

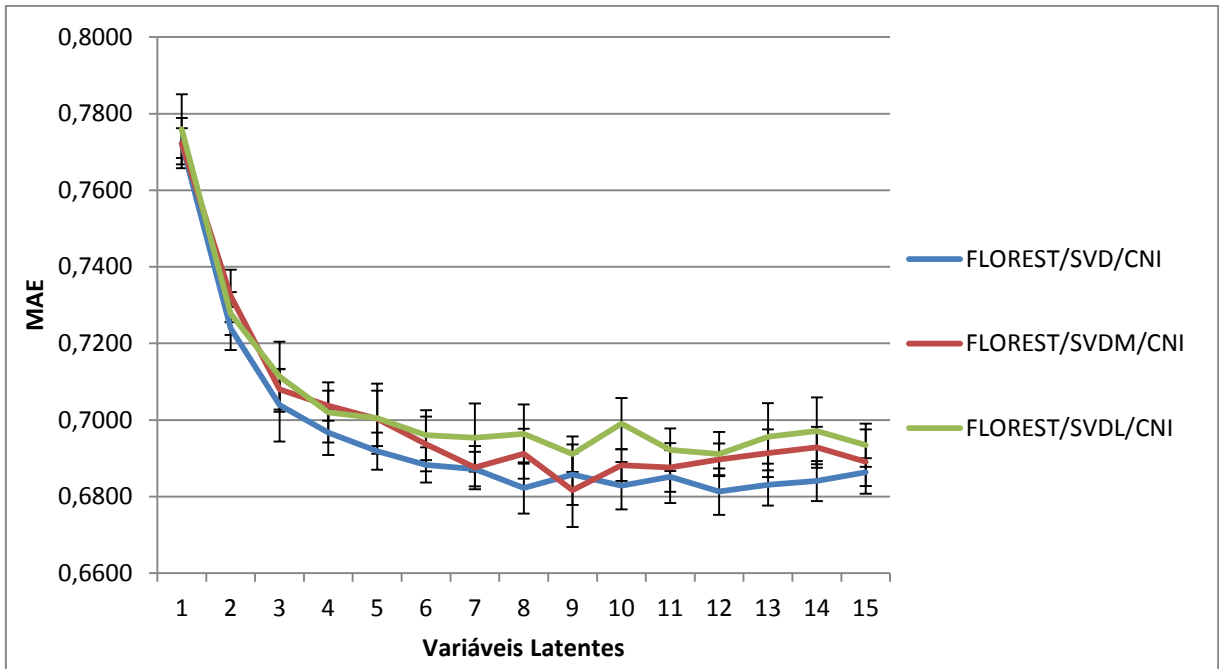


Figura 46 - MAE do experimento utilizando Floresta Aleatória e com normalização através da média do item.

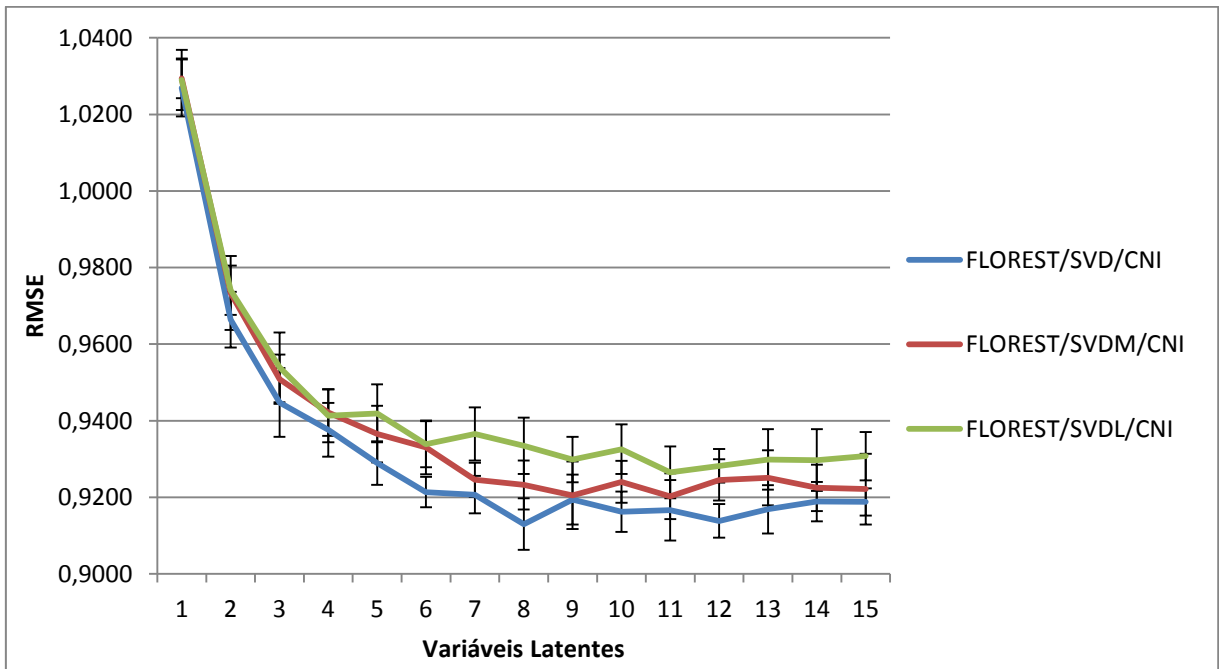


Figura 47 - RMSE do experimento utilizando Floresta Aleatória e com normalização através da média do item.

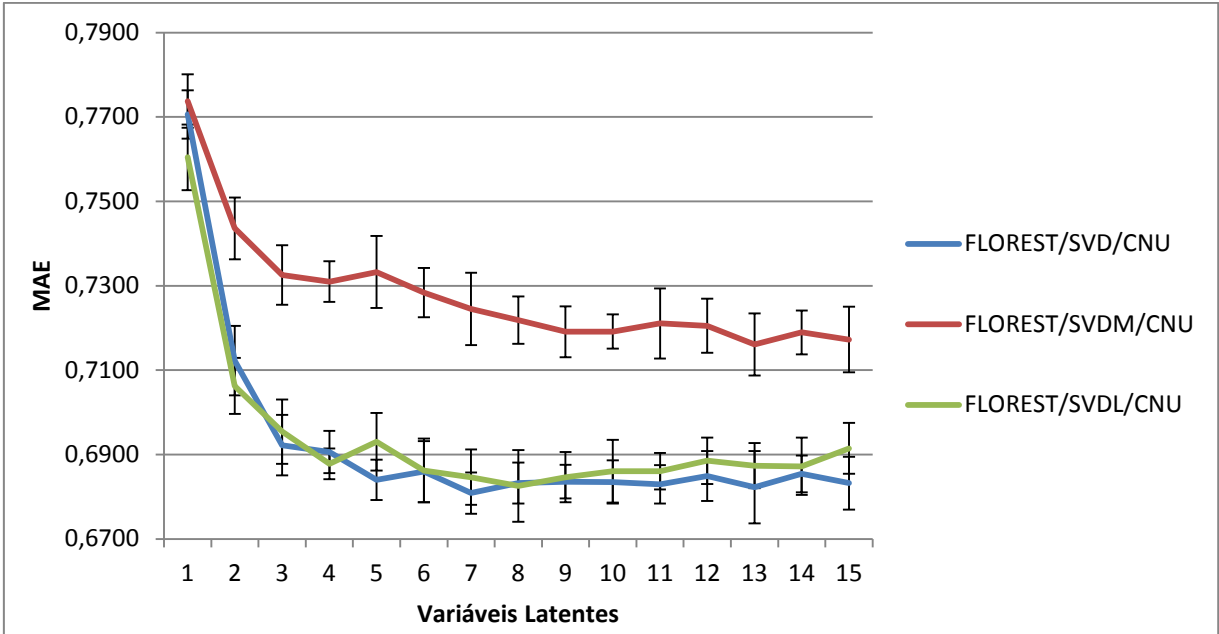


Figura 48 - MAE do experimento utilizando Floresta Aleatória e com normalização através da média do usuário.

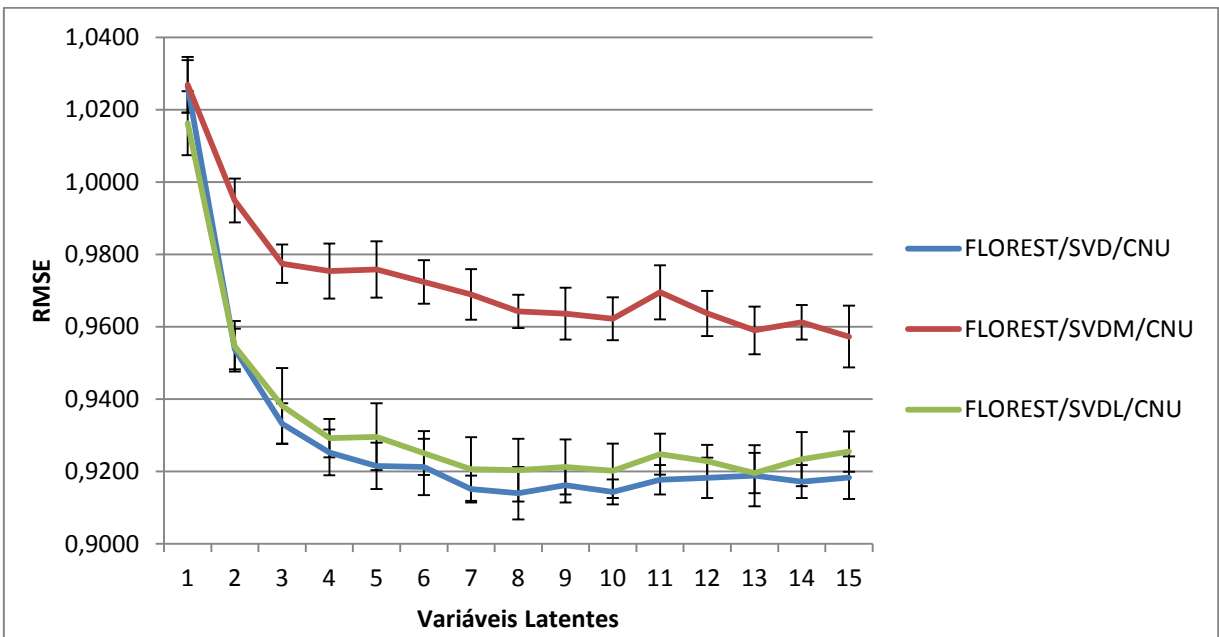


Figura 49 - RMSE do experimento utilizando Floresta Aleatória e com normalização através da média do usuário.

O experimento obteve características diferentes para cada algoritmo utilizado. No caso da utilização da técnica aprendizado bayesiano verificou que o aumento do número de variáveis latentes teve como consequência o aumento do erro tanto para o MAE quanto para o RMSE. No caso da rede neural obteve uma forte queda do erro, mas ao aumentar muito o número de variáveis latentes a técnica começa obter um resultado pior. Nesse caso não existiu um padrão claro com relação a normalização e a técnica de extração das variáveis latentes, mas na maioria dos casos o *Partial SVD* utilizando alguma normalização obteve um resultado melhor.

No último a floresta aleatória teve uma tendência de queda quando aumenta o número de variáveis latentes até uma estabilização desta. De modo geral, os erros variando as técnicas de extração de variáveis latentes e a técnica SVD obtendo um erro melhor. Somente na normalização por usuário que o *Partial SVD* obteve um erro muito maior que as outras técnicas. No experimento I utilizamos o valor de oito variáveis latentes como base dos experimentos. Verificou que no caso da floresta aleatória obteve o melhor resultado por volta desse valor.

4.4.3 – Experimento III

O experimento anterior foi responsável por verificar o desempenho permutando os parâmetros definidos e descobrir aquela que tenha a melhor configuração. Foi detectada que a melhor configuração é a utilização da técnica SVD para extração das variáveis latentes, normalização por usuário, oito variáveis latentes para usuário e para o item e floresta aleatória como algoritmo de aprendizado supervisionado.

Esse experimento é responsável por comparar o melhor resultado do experimento anterior (floresta aleatória, utilizando SVD e oito variáveis latentes) com as técnicas clássicas da filtragem colaborativa. Foi comparado com quatro técnicas da literatura da filtragem colaborativa: vizinhos mais próximos de um usuário e de um item e para cada utilizando a similaridade por cosseno e por Pearson, *SVD by Funk* e o *SVD Model*. Segue abaixo os resultados encontrados tanto para o erro quanto a cobertura dos algoritmos, ou seja, a porcentagem das instâncias que o algoritmo conseguiu avaliar.

1. Algoritmos Baseados em Memória

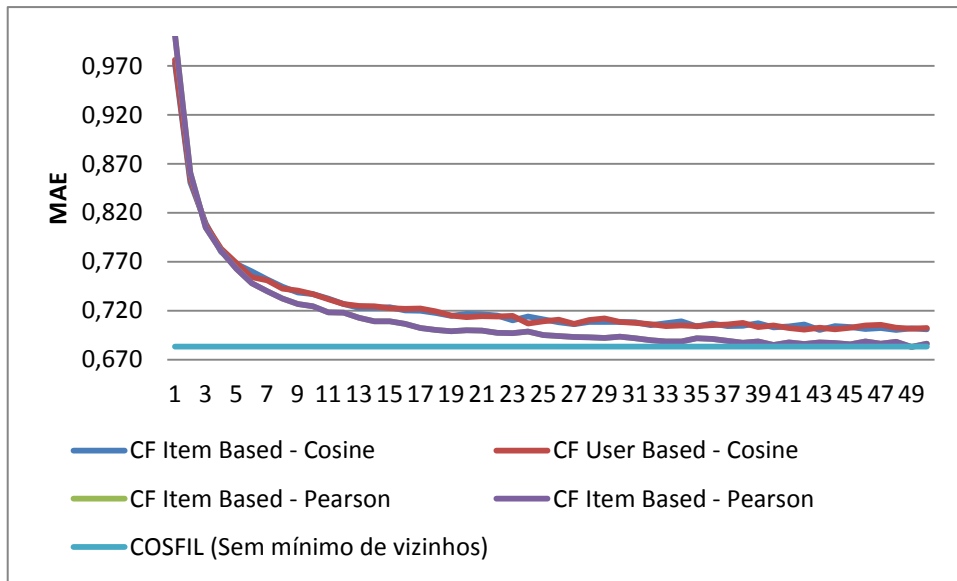


Figura 50 - MAE dos algoritmos baseados em memória e comparando com o melhor configuração da proposta.

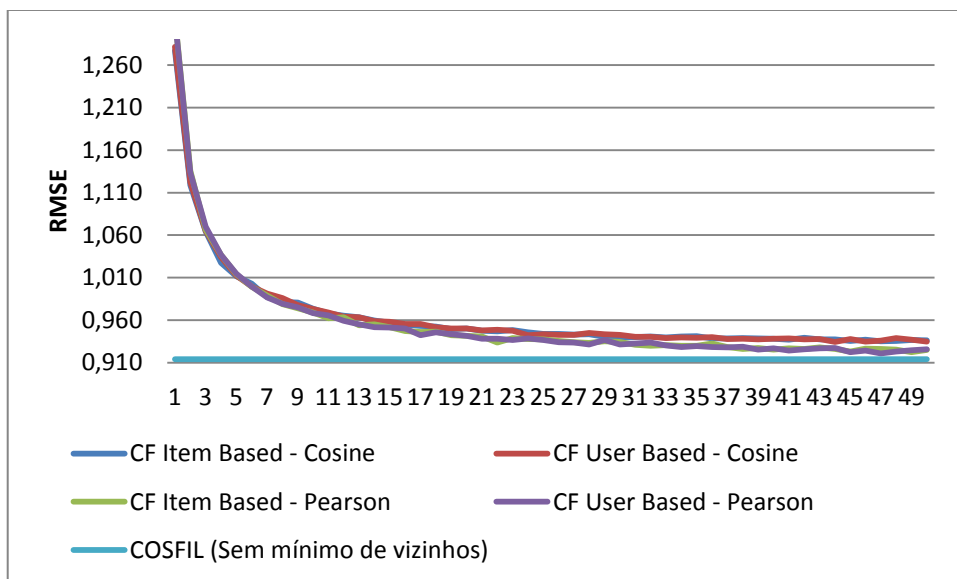


Figura 51 - RMSE dos algoritmos baseados em memória e comparando com o melhor configuração da proposta.

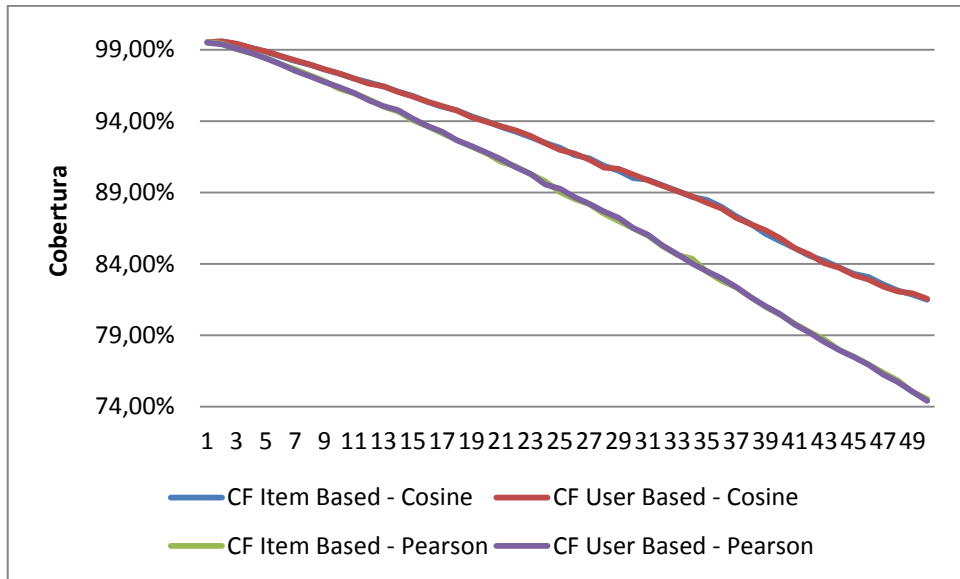


Figura 52 – Cobertura dos algoritmos baseados em memória.

2. Algoritmos baseados em Modelos

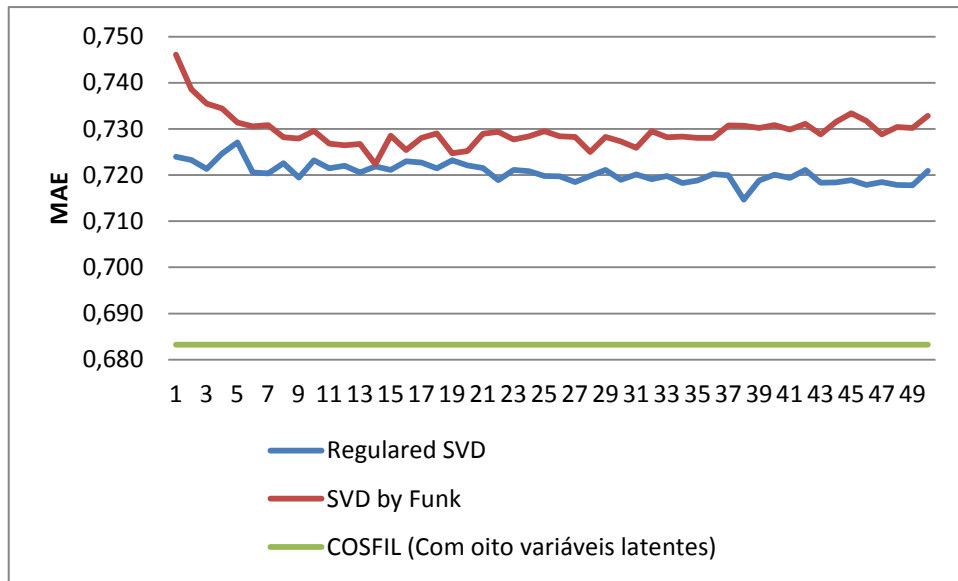


Figura 53 - MAE dos algoritmos baseados em modelo e comparando com o melhor configuração da proposta.

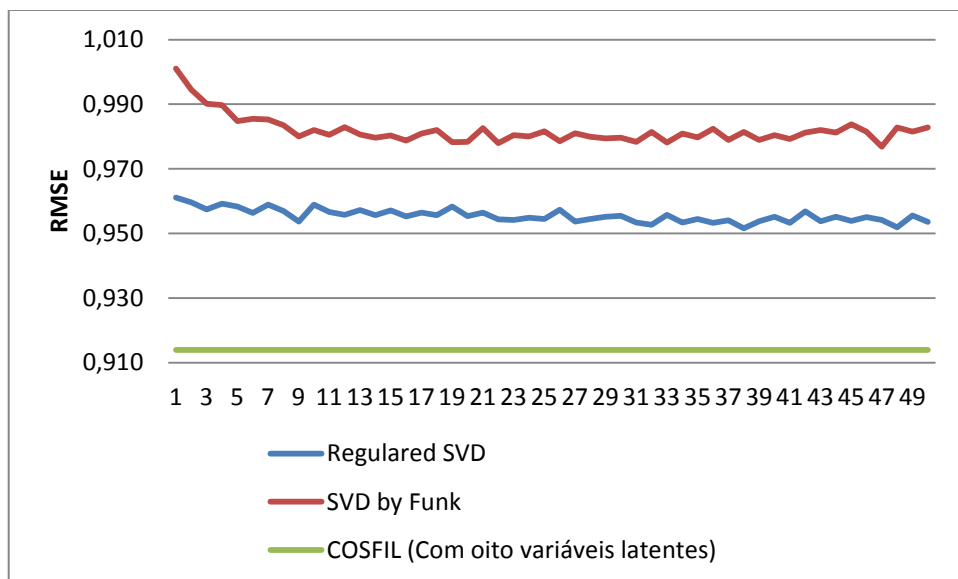


Figura 54 - RMSE dos algoritmos baseados em modelo e comparando com o melhor configuração da proposta.

4.4.4 – Experimento IV

Os algoritmos de vizinhos mais próximos não possuem 100% de cobertura da base, dado que determinado usuários ou itens não possuem o número mínimo de vizinhos. No entanto, os métodos com abordagem utilizando modelo não possuem essa limitação apresentando 100% de cobertura. Porém, para uma comparação mais justa, convém investigar

o que ocorre quando diminuimos a cobertura da nossa proposta variando o número mínimo de vizinhos.

Nesse experimento é avaliado o desempenho da melhor configuração obtida no experimento II (floresta aleatória, utilizando SVD e oito variáveis latentes) variando o número mínimo de vizinhos necessários para realizar a previsão. O objetivo do experimento é poder comparar o desempenho da proposta com as técnicas de vizinhos mais próximos que possuem essa característica e verificar a influência no desempenho quando é definido o limite mínimo de vizinhos. Seguem os resultados encontrados.

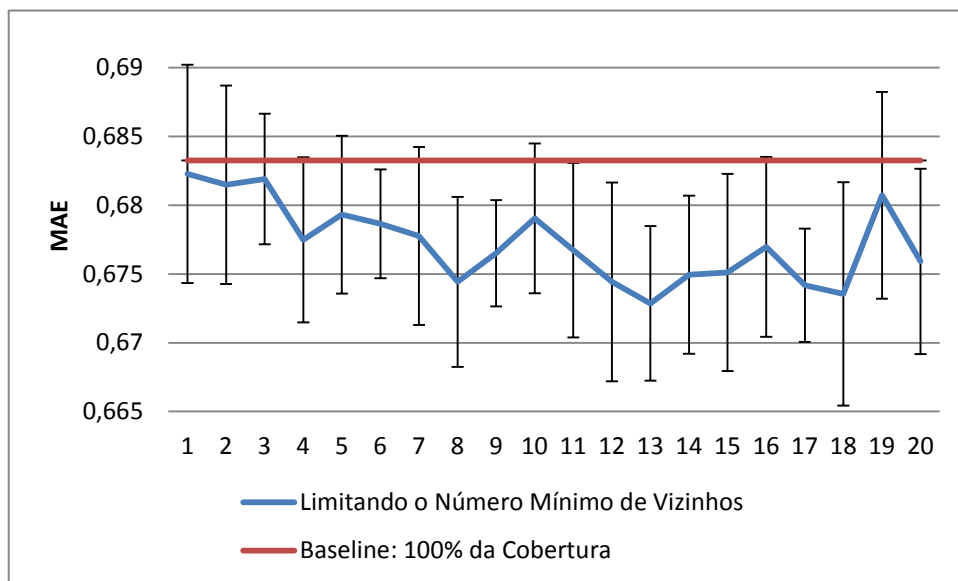


Figura 55 - MAE do experimento utilizando a melhor configuração do experimento II variando o número de vizinhos.

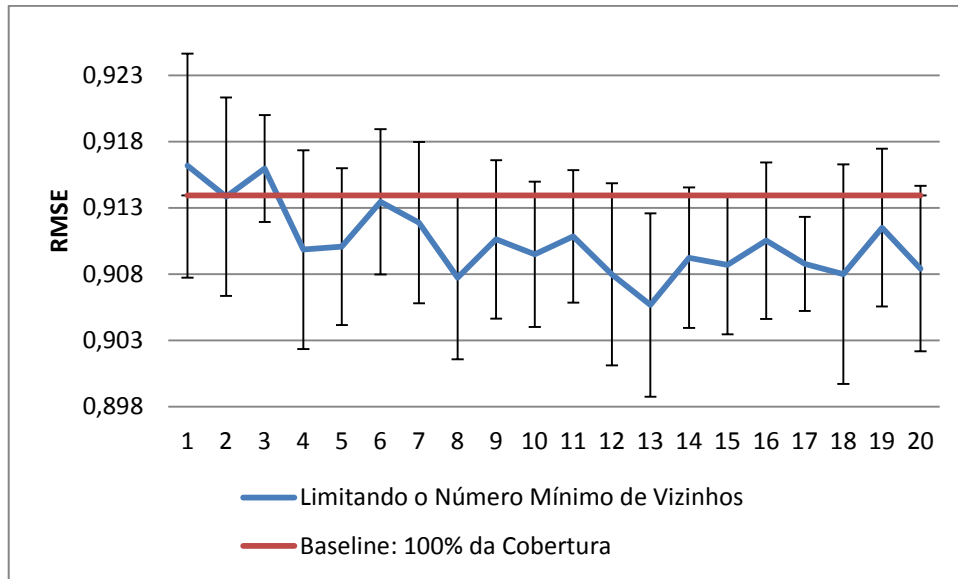


Figura 56 - RMSE do experimento utilizando a melhor configuração do experimento II variando o número de vizinhos.

4.5 - Análise dos Dados

Na presente seção, serão discutidos os resultados apresentados até o momento, evidenciando os resultados obtidos com a variação das configurações da técnica proposta. Por fim, serão comparados os resultados das técnicas clássicas de sistema de recomendação com a melhor configuração da proposta.

O segundo experimento tinha o objetivo em variar as configurações possíveis da proposta, sendo que os parâmetros dos algoritmos de aprendizado de máquina foram definidos no primeiro. Basicamente houve dois comportamentos distintos desses resultados. O primeiro é quando aumenta o número de característica também melhora o desempenho, ou melhor, quando aumenta o conhecimento sobre os dados há uma melhora no desempenho até um limite (redes neurais e floresta aleatória) e a outra ao contrário, ou seja, piorando o resultado (aprendizado bayesiano).

A Tabela 5 contém os melhores resultados do experimento II, separando o melhor resultado para o par algoritmo de aprendizado supervisionado e normalização. Por causa da escolha de utilizar duas métricas para a avaliação do desempenho, sendo que não necessariamente o melhor em uma é melhor na outra, foi apresentada duas colunas onde será informado o melhor resultado com relação a cada métrica.

Tabela 5 – Os melhores resultados (MAE e RMSE) do experimento dois e seus respectivas configurações.

		Melhor MAE				Melhor RMSE			
Técnica	Normalização	MAE	RMSE	V.L.	T.E.V.L.	MAE	RMSE	V.L.	T.E.V.L.
N. Bayes	Sem N.	0.729	1.050	1	SVDL	0.729	1.050	1	SVDL
ANN	Sem N.	0.691	0.930	2	SVDL	0.691	0.930	2	SVDL
	Usuário	0.689	0.932	3	SVDL	0.689	0.932	3	SVDL
	Item	0.700	0.944	2	SVDL	0.703	0.942	6	SVD
Floresta A.	Sem N.	0.687	0.930	6	SVDL	0.690	0.918	9	SVDL
	Usuário	0.681	0.915	7	SVD	0.683	0.914	8	SVD
	Item	0.681	0.914	12	SVD	0.682	0.913	8	SVD

Como se pode verificar nos resultados obtidos a escolha da representação dos dados com variáveis latentes influenciou de forma diferenciada para cada algoritmo de aprendizado supervisionado, evidenciando a hipótese levantada de que a escolha da técnica da extração das variáveis latentes influencia diferentemente cada algoritmo. De modo geral, o *Partial SVD* gerou o melhor resultado para o aprendizado bayesiano e rede neural, e o SVD puro na floresta aleatória.

Além da técnica de aprendizado de máquina, a quantidade de variáveis latentes afetou o resultado de modo diferenciado dependendo do algoritmo escolhido. O aprendizado bayesiano obteve um melhor resultado com poucas variáveis. Já a rede neural teve um resultado melhor com quatro e a floresta aleatória com oito variáveis. Quando o melhor resultado é utilizando a técnica SVD para extração das variáveis latentes, os melhores resultados deram em torno de oito variáveis. A mesma quantidade de variáveis onde possui o último salto significativo nos valores singulares, como pode ser visto na e Figura 30 e na Figura 31.

Durante a apresentação da proposta aventamos a hipótese de que a forma de completar os valores faltantes da matriz usuário/item impactaria diretamente os resultados obtidos, seja utilizando zero ou alguma média. Como pode ser visto nos resultados, a diferença entre utilizar a normalização (média do usuário ou do item) contra a forma mais simples de representar os dados (utilizando zero nos valores desconhecidos) não gerou uma grande diferença nos resultados: na média a diferença ficou abaixo de 1%.

O experimento III foi responsável por avaliar as técnicas clássicas da filtragem colaborativa utilizando a mesma metodologia que foi aplicada na avaliação dos experimentos

da nossa proposta. A técnica *Regulated SVD* obteve um resultado melhor comparado com as técnicas de filtragem colaborativa baseadas em memória e a *SVD by Funk*, mas quando é determinado um limite no número mínimo de vizinhos nas técnicas baseadas em memória, essas obtêm um bom resultado, e até melhores do que os baseados em modelo. Um problema dessa limitação é a diminuição da cobertura.

Os resultados obtidos usando a nossa proposta foram consistentemente melhores que os obtidos com os algoritmos clássicos da filtragem colaborativa. Nossa proposta gerou uma melhora de aproximadamente 5% no MAE e 4% no RMSE comparado com o melhor algoritmo da filtragem colaborativa. Mesmo limitando em 50 o número de vizinhos nas técnicas baseadas em memória, a proposta obteve um resultado melhor e com cobertura total da base de teste. Esses resultados podem ser visto na Tabela 6.

Tabela 6 - Comparação da proposta (COFISL) com as técnicas clássicas da área de sistema de recomendação.

Técnica	MAE	RMSE	Cobertura	
COFISL	0.681	0.914	100%	
COFISL	0.673	0.906	96.40%	Min.13 Vizinhos
CF. Item Based Cosine	0.723	0.963	96.42%	Min.13 Vizinhos
CF. User Based Cosine	0.725	0.963	96.45%	Min.13 Vizinhos
CF. Item Based Pearson	0.713	0.954	95.01%	Min.13 Vizinhos
CF. User Based Pearson	0.715	0.955	95.05%	Min.13 Vizinhos
CF. Item Based Cosine	0.700	0.936	82.14%	Min.48 Vizinhos
CF. User Based Cosine	0.700	0.938	84.65%	Min.42 Vizinhos
CF. Item Based Pearson	0.683	0.922	75.05%	Min.49 Vizinhos
CF. User Based Pearson	0.682	0.924	76.95%	Min.46 Vizinhos
SVD by Funk	0.722	0.980	100%	14 Variáveis
Regulated SVD	0.715	0.952	100%	38 Variáveis

Limitando o número mínimo de vizinho no algoritmo proposto houve uma queda de aproximadamente de 2% no MAE e 1% no RMSE. O melhor resultado da proposta ocorreu quando a limitação do número de vizinhos chegou ao valor de treze. Definindo essa quantidade mínima como parâmetro, a proposta obteve uma melhoria em comparado com o melhor resultado dos algoritmos baseados em memória de aproximadamente de 6% no MAE e 5% no RMSE.

Além disso, o melhor algoritmo de modelo, *Regularized SVD*, gerou o melhor resultado quando usou 38 variáveis latentes enquanto a nossa proposta já obteve um resultado melhor com apenas duas. A complexidade do algoritmo SVD é diretamente proporcional ao número de variáveis latentes que se deseja calcular, logo nossa proposta reduz o tempo necessário de pré-processamento (SVD).

Somente o MAE ou RMSE não é o suficiente para entender com plenitude como a recomendação está sendo realizada. Para tal, foi calculada a matriz de confusão do algoritmo de filtragem colaborativa baseado no usuário utilizando similaridade cosseno e a melhor configuração da proposta. Essa tabela permite visualizar o desempenho de um algoritmo supervisionado. Cada coluna representa as instâncias de uma classe prevista, enquanto as linhas as instâncias de classe real. A Tabela 7 representa a filtragem colaborativa baseado no usuário e na Tabela 8 o resultado da proposta.

Tabela 7 – Filtragem Colaborativa baseado no usuário utilizando a similaridade cosseno e o mínimo de treze vizinhos.

		Classe Prevista					Total	Porcentagem
		1	2	3	4	5		
Classe Real	1	45	170	231	60	2	508	8,86%
	2	11	204	612	249	13	1089	18,73%
	3	8	196	1369	988	43	2604	52,57%
	4	1	75	1192	1951	141	3360	58,07%
	5	2	23	389	1392	283	2089	13,55%
Total		67	668	3793	4640	482	9650	
Porcentagem		67,16 %	30,54 %	36,09 %	42,05 %	58,71 %		

Tabela 8 – Matriz de confusão da proposta utilizando o algoritmo floresta aleatória, SVD, normalização através da média do usuário e oito variáveis latentes.

		Classe Prevista					Total	Porcentagem
		1	2	3	4	5		
Classe Real	1	53	195	266	73	3	590	8,98%
	2	9	216	683	244	8	1160	18,62%
	3	1	183	1440	1052	21	2697	53,39%
	4	0	49	1049	2190	143	3431	63,83%
	5	2	16	298	1429	372	2117	17,57%
Total		65	659	3736	4988	547	9995	
Porcentagem		81,54%	32,78%	38,54%	43,91%	68,01%		

Uma das maiores dificuldades na recomendação, dentro da filtragem colaborativa, é prever uma nota baixa. Os algoritmos possuem uma tendência de ter um erro alto nesses casos, pois tendem a classificar com a nota média. Através da matriz de confusão (Tabela 7 e Tabela 8) pode-se ver que as duas técnicas erram bastante nesse quesito. Nos dois casos tendem a recomendar a nota três, ou seja, a nota média dessa base de dados. Uma grande surpresa foi no caso da proposta quando o algoritmo informa que a recomendação é a nota um, a probabilidade do algoritmo estar correto é alta (81,54%) e muito superior da técnica dos vizinhos mais próximos.

De modo geral, a proposta obteve um resultado melhor do que o algoritmo dos vizinhos mais próximos mesmo prevendo mais avaliações. Nas notas quatro e cinco houve um desempenho superior da proposta em comparação com o algoritmo clássico. Outra melhoria foi na redução das previsões extremas, ou seja, quando a nota é um valor e o algoritmo prevê a pior nota possível, por exemplo, a nota é cinco e ele prevê um. A única desvantagem foi no caso da nota um. O algoritmo proposto classificou com nota mais extrema nesse caso comparada com o clássico.

Através desses resultados pode-se concluir que existe uma diminuição considerável tanto no MAE quanto no RMSE utilizando a proposta comparada com os algoritmos clássicos da filtragem colaborativa. Sendo que essa comparação foi feita nos algoritmos clássicos dentro das duas classes mais utilizadas na filtragem colaborativa: os baseados em memória e em modelo. A redução do MAE/RMSE foi porque o algoritmo conseguiu generalizar melhor o problema, como foi visto na matriz de confusão.

Capítulo 5 – Conclusões

Este capítulo aborda as considerações finais referentes a este trabalho, suas contribuições e perspectivas futuras.

5.1 Considerações acerca do trabalho

Este trabalho propôs uma transformação do problema de filtragem colaborativa em um problema de aprendizado supervisionado de máquina, levantando a hipótese que é possível realizar essa transformação e assim utilizar os algoritmos de aprendizado supervisionado para a previsão dos valores desconhecidos.

A filtragem colaborativa, uma das abordagens de recomendação, utiliza as preferências explícitas dos usuários sobre os itens para a tarefa de recomendar. Contudo, mesmo possuindo características de um típico problema de reconhecimento de padrões, não é possível construir diretamente um conjunto de treinamento por causa da conjectura do problema e assim aplicar um algoritmo de aprendizado supervisionado.

Para tal, esse trabalho propôs um *framework* para a aplicação das técnicas de aprendizado supervisionado, e através dela, estudá-lo como um problema típico de aprendizado supervisionado. Por fim, foram realizados experimentos utilizando a proposta e seus resultados apresentados neste trabalho. Em função disso, foi efetuada uma análise dos resultados comparando com os algoritmos clássicos da literatura, e os nossos resultados foram melhores.

5.2 Contribuições

A principal contribuição deste trabalho é a validação da hipótese de que é possível transformar o problema de filtragem colaborativa em um problema de aprendizado supervisionado. Para tal, foi proposto um *framework* que realiza essa transformação e partir daí tratar o problema da filtragem colaborativa como um de aprendizado supervisionado. Foi levantada a suposição que existem variáveis latentes que representem os usuários e os itens, e com isso deve ser possível construir um espaço onde podemos representar as notas através dessas variáveis e assim aplicar os algoritmos de aprendizado supervisionado.

Com o objetivo de validar a proposta foram realizados diversos experimentos visando à avaliação do desempenho da técnica utilizando diversas combinações dos parâmetros, como

exemplo, o método utilizado para extração das variáveis latentes, o algoritmo de aprendizado supervisionado, número de variáveis e outros, e por fim foi comparado o desempenho da proposta com as técnicas clássicas da literatura de sistema de recomendação.

Os experimentos realizados utilizando a transformação obtiveram resultados satisfatórios e superiores aos algoritmos clássicos. Além de ser superior com relação às métricas, como MAE e o RMSE, a proposta obteve uma redução na previsão extremas, ou seja, quando a nota é um valor e o algoritmo prever a pior nota possível, por exemplo, a nota é cinco e ele prevê um. Esse tipo de nota é fundamental em um sistema de recomendação, pois é pior recomendar um item que o usuário não goste do que um item que seja mediano para ele.

5.3 Limitações e trabalhos futuros

Esta seção descreve potenciais pontos de melhoria, ampliação e aprimoramento da transformação do problema de filtragem colaborativa para aprendizado de máquina supervisionada e experimentos futuros, alguns dos quais são também soluções às limitações também descritas.

Os experimentos mostram que a utilização da proposta para a previsão de notas dentro do problema de filtragem colaborativa é válida, ou seja, produz melhores resultados que os métodos tradicionais. Neste trabalho, foram realizados experimentos que demonstraram a eficiência da aplicação dos algoritmos da área de aprendizado de máquina supervisionado comparado com as técnicas clássicas da filtragem colaborativa. Portanto, é possível tratar esse problema como um problema de reconhecimento de padrões e utilizar as técnicas desta área para melhorar o seu desempenho.

Um desafio é utilizar tal abordagem em um sistema de recomendação real. Todos os experimentos foram realizados visando um sistema de recomendação *off-line*. Contudo, não basta ter uma boa taxa de acerto, o algoritmo precisa ser escalável. Um dos grandes problemas dos algoritmos supervisionados mais elaborados é o tempo no treinamento do modelo. Existem diversas possibilidades para solucionar esse problema como a utilização de técnicas apropriadas que conseguem convergir um modelo rapidamente com uma grande quantidade de dados (HUANG *et al.*, 2004, 2011), modelos com treinamento incremental (JOSHI, 2012)

e/ou técnicas para a seleção dos melhores dados para serem utilizados para o treinamento (MELLO *et al.*, 2010, SUN & WANG, 2010).

Outra vantagem de utilizar os algoritmos supervisionados é a sua capacidade de generalização. Em um sistema de recomendação real utilizando essa abordagem poderia treinar um modelo utilizando um conjunto pequeno, parcial e representativo dos dados e que seja capaz do modelo generalizar, e assim reduzir o tempo de treinamento. Com isso o sistema seria pouco afetado quando surgem notas novas, precisando somente gerar um novo modelo ou adicionar esse conhecimento ao antigo quando existir um conjunto de novos dados o suficiente para melhorar a generalização.

Um problema clássico da filtragem colaborativa é o problema do *cold-start*. Essa abordagem sofre desse problema, porque não teria como representar um item ou usuário devido a sua falta de avaliações e os seus atributos no novo espaço de variáveis seriam nulos. Um trabalho futuro seria estudar como tratar esses casos, como exemplo, no caso do usuário novo, recomendar aleatoriamente um item, recomendar os itens melhores avaliados na base e outros, e no caso de um item, recomendar de vez em quando para um usuário um item novo ou pouco avaliado.

Uma limitação do trabalho é a utilização de uma única base para testar e validar o algoritmo. Dentro da literatura existem diversas bases diferentes e cada uma possuem características diferentes, como o tamanho do conjunto de preferência, a distribuição das notas tanto para o usuário e para o item, a quantidade de avaliações, o grau de esparsidade e outros. Dependendo dessas características o algoritmo proposto pode obter um melhor ou um pior resultado. Um trabalho futuro seria avaliar a proposta nessas outras bases de dados que utilizam como base a filtragem colaborativa.

Referências Bibliográficas

- ADOMAVICIUS, G., TUZHILIN, A., 2005, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions". In: *IEEE Transactions on Knowledge and Data Engineering*. v. 17, n. 6 (Jun.), pp. 734–749.
- ANSARI, A., ESSEGAIER, S., KOHLI, R., 2000, "Internet recommendation systems". In: *Journal of Marketing research*.
- BAEZA-YATES, R., RIBEIRO-NETO, B., 1999, *Modern information retrieval*. S.l., Addison-Wesley New York.
- BALABANOVIĆ, M., SHOHAM, Y., 1997, "Fab: content-based, collaborative recommendation". In: *Communications of the ACM*. v. 40, n. 3, pp. 66–72.
- BELLMAN, R.E., 1961, *Adaptive control processes - A guided tour*. Princeton, New Jersey, U.S.A., Princeton University Press.
- BILLSUS, D., 1998, "Learning collaborative information filters". In: *International Conference on Machine Learning*.
- BILLSUS, D., PAZZANI, M.J., 2000, "User modeling for adaptive news access". In: *User modeling and user-adapted interaction*. v. 10, n. 2, pp. 147–180.
- BISHOP, C.M., 2006, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. S.l., Springer-Verlag New York, Inc.
- BREESE, J.S., HECKERMAN, D., KADIE, C., et al., 1998. "Empirical analysis of predictive algorithms for collaborative filtering". In: *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*. S.l.: s.n. 1998. pp. 43–52.
- BREIMAN, L., 1984, "Classification and regression trees". In: HALL CRC, Chapman (ed.), *The Wadsworth statisticsprobability series*. v. 19, n. Book, Whole, pp. 368.
- BURKE, R., 2007. "Hybrid web recommender systems". In: *The adaptive web*. S.l.: Springer-Verlag. 2007. pp. 377–408.
- CANDÈS, E.J., RECHT, B., 2009, "Exact matrix completion via convex optimization". In: *Foundations of Computational Mathematics*. v. 9, n. 6 (Apr.), pp. 717–772.

- CANNY, J., 2002. "Collaborative filtering with privacy via factor analysis". In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02*. New York, New York, USA: ACM Press. 2002. pp. 238.
- CARUANA, R., KARAMPATZIAKIS, N., 2008, "An empirical evaluation of supervised learning in high dimensions". In: *on Machine learning*. pp. 96–103.
- CHEN, Y.H., GEORGE, E.I., 1999. "A bayesian model for collaborative filtering". In: *Proc of the 7th International Workshop on Artificial Intelligence and Statistics*. S.l.: s.n. 1999.
- CLAYPOOL, M., GOKHALE, A., MIRANDA, T., et al., 1999. "Combining content-based and collaborative filters in an online newspaper". In: *Proceedings of ACM SIGIR Workshop on Recommender Systems*. S.l.: Citeseer. 1999. pp. 60.
- CONDLIFF, M., LEWIS, D., MADIGAN, D., et al., 1999, "Bayesian mixed-effects models for recommender systems". In: *Proc. ACM SIGIR*.
- CUNNINGHAM, P., SMYTH, B., 2010, "An assessment of machine learning techniques for review recommendation". In: *Artificial Intelligence and Cognitive Science*. pp. 241–250.
- CYBENKO, G., 1989, "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals, and Systems (MCSS)*. pp. 303–314.
- DEERWESTER, S., DUMAIS, S.T., FURNAS, G.W., et al., 1990, "Indexing by latent semantic analysis". In: *Journal of the American society for information science*. v. 41, n. 6, pp. 391–407.
- DEMPSTER, A.P., LAIRD, N.M., RUBIN, D.B., 1977, "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)*. v. 39, n. 1, pp. 1–38.
- DUMAIS, S.T., 2005, "Latent semantic analysis". In: *Annual Review of Information Science and Technology*. v. 38, n. 1 (Sep.), pp. 188–230.
- FUNK, S., 2012. Disponível em: <<http://sifter.org/~simon/journal/20061211.html>>. Acessado em: 19 Agosto 2012.

- GE, M., DELGADO-BATTENFELD, C., JANNACH, D., 2010. "Beyond accuracy: evaluating recommender systems by coverage and serendipity". In: *Proceedings of the fourth ACM conference on Recommender systems*. S.l.: ACM. 2010. pp. 257–260.
- GOLDBERG, D., NICHOLS, D., OKI, B.M., et al., 1992, "Using collaborative filtering to weave an information tapestry". In: *Communications of the ACM*. v. 35, n. 12, pp. 61–70.
- GOLDBERG, K., ROEDER, T., GUPTA, D., et al., 2001, "Eigentaste: A constant time collaborative filtering algorithm". In: *Information Retrieval*. v. 4, n. 2, pp. 133–151.
- HERLOCKER, J.L., KONSTAN, J. A., TERVEEN, L.G., et al., 2004, "Evaluating collaborative filtering recommender systems". In: *ACM Transactions on Information Systems*. v. 22, n. 1 (Jan.), pp. 5–53.
- HO, T., 1995, "Random decision forests". In: *Analysis and Recognition, 1995., Proceedings of the*.
- HOFMANN, T., 2003. "Collaborative filtering via gaussian probabilistic latent semantic analysis". In: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. S.l.: ACM. 2003. pp. 259–266.
- HOFMANN, T., 2004, "Latent semantic models for collaborative filtering". In: *ACM Transactions on Information Systems*. v. 22, n. 1 (Jan.), pp. 89–115.
- HSU, S., WEN, M.H., LIN, H.C., et al., 2007, "Aimed-a personalized tv recommendation system". In: *Interactive TV: a Shared Experience*. pp. 166–174.
- HUANG, C.B., GONG, S.J., 2008. "Employing rough set theory to alleviate the sparsity issue in recommender system". In: *Machine Learning and Cybernetics, 2008 International Conference on*. S.l.: IEEE. 2008. pp. 1610–1614.
- HUANG, G., ZHU, Q., SIEW, C., 2004, "Extreme learning machine: a new learning scheme of feedforward neural networks". In: *Neural Networks, 2004.* v. 2, pp. 985–990.
- HUANG, G.-B., WANG, D.H., LAN, Y., 2011, "Extreme learning machines: a survey". In: *International Journal of Machine Learning and Cybernetics*. v. 2, n. 2 (May), pp. 107–122.
- HUNT, E., MARIN, J., STONE, P., 1966, *Experiments in induction*. S.l., Academic Press.

- JOSHI, P., 2012, "Incremental Learning: Areas and Methods – A Survey". In: *International Journal of Data Mining & Knowledge Management Process*. v. 2, n. 5 (Sep.), pp. 43–51.
- KOREN, Y., BELL, R., VOLINSKY, C., 2009, "Matrix factorization techniques for recommender systems". In: *Computer*. v. 42, n. 8, pp. 30–37.
- LINDEN, G., SMITH, B., YORK, J., 2003, "Amazon. com recommendations: Item-to-item collaborative filtering". In: *Internet Computing, IEEE*. v. 7, n. 1, pp. 76–80.
- MARSLAND, S., 2009, *Machine learning: an algorithmic perspective*. S.l., Chapman & Hall/CRC.
- MCCULLOCH, W., PITTS, W., 1943, "A logical calculus of the ideas immanent in nervous activity". In: *Bulletin of mathematical biology*. v. 5, pp. 115–133.
- MELLO, C.E., AUFAURE, M.A., ZIMBRAO, G., 2010. "Active learning driven by rating impact analysis". In: *Proceedings of the fourth ACM conference on Recommender systems*. S.l.: ACM. 2010. pp. 341–344.
- MELVILLE, P., MOONEY, R.J., NAGARAJAN, R., 2002. "Content-boosted collaborative filtering for improved recommendations". In: *Proceedings of the National Conference on Artificial Intelligence*. S.l.: Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. 2002. pp. 187–192.
- MILLER, B., ALBERT, I., LAM, S., 2003, "MovieLens unplugged: experiences with an occasionally connected recommender system". In: *Proceedings of the 8th* pp. 263–266.
- MITCHELL, T.M., 1997a, *Machine Learning*. 1. S.l., McGraw-Hill Science Engineering.
- MITCHELL, T.M., 1997b, *Machine Learning*. 1. S.l., McGraw-Hill Science Engineering.
- O'MAHONY, M.P., SMYTH, B., 2009. "Learning to recommend helpful hotel reviews". In: *Proceedings of the third ACM conference on Recommender systems*. S.l.: ACM. 2009. pp. 305–308.
- O'MAHONY, M.P., SMYTH, B., 2010. "Using readability tests to predict helpful product reviews". In: *Adaptivity, Personalization and Fusion of Heterogeneous Information*. S.l.: LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE. 2010. pp. 164–167.

- PAPAGELIS, M., PLEXOUSAKIS, D., KUTSURAS, T., 2005, "Alleviating the sparsity problem of collaborative filtering using trust inferences". In: *Trust Management*. pp. 125–140.
- PATEREK, A., 2007. "Improving regularized singular value decomposition for collaborative filtering". In: *Proceedings of KDD Cup and Workshop*. S.l.: s.n. 2007. pp. 5–8.
- PAZZANI, M.J., 1999, "A framework for collaborative, content-based and demographic filtering". In: *Artificial Intelligence Review*. v. 13, n. 5, pp. 393–408.
- POPESCU, A., PENNOCK, D., LAWRENCE, S., 2001, "Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments". In: *Proceedings of the Seventeenth ...*. pp. 437–444.
- QUINLAN, J., 1986, "Induction of decision trees". In: *Machine learning*. pp. 81–106.
- QUINLAN, J., 1993, *C4. 5: programs for machine learning*. S.l., Morgan Kaufmann Publishers Inc.
- RICCI, F., ROKACH, L., SHAPIRA, B., et al., 2011, "Recommender Systems Handbook". In: RICCI, Francesco, ROKACH, Lior, SHAPIRA, Bracha & KANTOR, Paul B. (eds.), *Media*.
- RISH, I., 2001, "An empirical study of the naive Bayes classifier". In: *IJCAI 2001 Workshop on Empirical Methods in Artificial*.
- ROSENBLATT, F., 1958, "The perceptron: a probabilistic model for information storage and organization in the brain.". In: *Psychological review*. v. 65, n. 6 (Nov.), pp. 386–408.
- SARWAR, B., KARYPIS, G., KONSTAN, J., et al., 2000, "Application of dimensionality reduction in recommender system-a case study". In: *Architecture*.
- SARWAR, B., KARYPIS, G., KONSTAN, J., et al., 2001. "Item-based collaborative filtering recommendation algorithms". In: *Proceedings of the 10th international conference on World Wide Web*. S.l.: ACM. 2001. pp. 285–295.
- SARWAR, B.M., KARYPIS, G., KONSTAN, J., et al., 2002. "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering". In: *Proceedings of the Fifth International Conference on Computer and Information Technology*. S.l.: s.n. 2002. pp. 158–167.

- SCHAFFER, J.B., KONSTAN, J., RIEDI, J., 1999. "Recommender systems in e-commerce". In: *Proceedings of the 1st ACM conference on Electronic commerce*. S.l.: ACM. 1999. pp. 158–166.
- SCHEIN, A.I., POPESCU, A., UNGAR, L.H., et al., 2002, "Methods and metrics for cold-start recommendations". In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02*. pp. 253.
- SHANI, G., HECKERMAN, D., BRAFMAN, R.I., 2006, "An MDP-based recommender system". In: *Journal of Machine Learning Research*. v. 6, n. 2, pp. 1265.
- SHARDANAND, U., MAES, P., 1995. "Social Information Filtering: Algorithms for Automating "Word of Mouth"". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. S.l.: ACM Press/Addison-Wesley Publishing Co. 1995. pp. 210–217.
- SHETH, B., 1993, "Evolving agents for personalized information filtering". In: *Artificial Intelligence for Applications, 1993*. pp. 345–352.
- SOBOROFF, I., NICHOLAS, C., 1999. "Combining content and collaboration in text filtering". In: *Proceedings of the IJCAI*. S.l.: s.n. 1999. pp. 86–91.
- SU, X., KHOSHGOFTAAR, T.M., 2009, "A Survey of Collaborative Filtering Techniques". In: *Advances in Artificial Intelligence*. v. 2009, n. Section 3, pp. 1–20.
- SUN, L., WANG, X., 2010, "A survey on active learning strategy". In: *Machine Learning and Cybernetics (ICMLC) n. July*, pp. 11–14.
- TAKÁCS, G., PILÁSZY, I., NÉMETH, B., et al., 2008. "Matrix factorization and neighbor based algorithms for the netflix prize problem". In: *Proceedings of the 2008 ACM conference on Recommender systems*. S.l.: ACM. 2008. pp. 267–274.
- TAN, P., STEINBACH, M., KUMAR, V., 2005, *Introduction to data mining*. S.l., Addison-Wesley Longman Publishing Co., Inc.
- TAPSCOTT, D., WILLIAMS, A.D., 2006, *Wikinomics: How Mass Collaboration Changes Everything*. S.l., Portfolio Hardcover.

TRAN, T., COHEN, R., 2000. "Hybrid recommender systems for electronic commerce". In: *Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press*. S.l.: s.n. 2000.

TREWIN, S., 2000, "Knowledge-based recommender systems". In: *Encyclopedia of Library and Information Science: pp. 1–23.*

WORLDWEBSIZE, 2012. Disponível em: <<http://www.worldwidewebsite.com/>>. Acessado em: 1 Outubro 2012.