Universidade Federal de Pernambuco

Centro de Informática

Curso de Bacharelado em Ciência da Computação

Teoria da Resposta ao Item para Avaliação de Algoritmos de Recomendação

Marcos da Silva Barreto

Recife
2019

Centro de Informática
Curso de Bacharelado em Ciência da Computação

Marcos da Silva Barreto

Teoria da Resposta ao Item para Avaliação de Algoritmos de Recomendação

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Ricardo Bastos C. Prudêncio

Recife
2019

Dedico esse trabalho aos meus pais,
por todo o amor e ensinamento que eles me deram,
e por sempre acreditar e apoiar meus sonhos e objetivos.
Tudo o que sou hoje devo a eles.

Agradecimentos

Em primeiro lugar, gostaria de agradecer primeiramente a minha família, em especial aos meus pais, Silberto e Maria do Socorro, e a minha irmã Camila, por todo o apoio carinho e amor em todas as minhas decisões, e no meu caminho nessa longa jornada.

A minha namorada Yone Kauane, por estar comigo, mesmo com a distância, em todos os meus momentos, sempre me ajudando a superar cada desafio, com muito carinho, amor e compreensão.

Aos meus amigos Leonardo, Lucas, Bruno, Vitor, Julius e Ananda, por me acompanharem desde o ensino médio, pelo apoio, e mesmo com a distância para alguns sempre mantivemos contato.

Aos meus professores do IFBA, em especial a Luzia Azevedo e Jonatas Bastos, por me encorajar e apoiar na decisão de vir para Recife e estudar no CIn - UFPE, assim como ajudar a encontrar uma casa para morar durante a graduação.

Ao professor Ricardo Prudêncio por me orientar neste trabalho de conclusão de curso, ao professor Paulo Salgado por ser meu avaliador, ao professor Paulo Borba por me orientar na minha primeira Iniciação Científica na faculdade, e a todos os docentes do CIn com quem tive a oportunidade de aprender.

Ao convênio CIn/Motorola que me deu a oportunidade do meu primeiro estágio e meu primeiro emprego, e a todas as pessoas que passaram por mim, me proporcionando novas amizades.

A Maria das Graças por tornar muito mais fácil a moradia em Recife, e me ajudar quando entrei para o CITi.

Resumo

Nos últimos anos o consumo de informação na internet aumentou significativamente, assim como a quantidade de usuários ativos na rede. Sistemas de recomendação têm sido amplamente utilizados por serviços web para direcionamento e sugestão de conteúdo, principalmente para filmes. No entanto, existem atualmente diversos algoritmos baseados em diferentes técnicas e abordagens, os quais apresentam melhores resultados em comparação com outros em determinados contextos. Este trabalho propõe um estudo da aplicação da Teoria da Resposta ao Item (TRI), um modelo estatístico utilizado para medir habilidades latentes de indivíduos baseado em suas respostas a um conjunto de problemas, como forma avaliativa para um grupo de 10 algoritmos de recomendação em contraste às métricas tradicionais. Para tal os sistemas de recomendação foram executados sobre a base de dados da MovieLens e avaliados pelo modelo B³-IRT, sendo analisados a habilidade latente dos algoritmos e os parâmetros de dificuldade e discriminação associados aos usuários. BaselineOnly e SVD alcançaram o menor RMSE (~0,85), porém os algoritmos KNNWithMeans_ItemBased e NMF obtiveram as maiores habilidades (~0,53), além disso, 44% dos usuários foram classificados com nível “médio” de dificuldade (~0,5). Os experimentos mostraram resultados diferentes dos esperados pelas métricas conhecidas, como o RMSE, porém o desempenho similar desses sistemas influenciaram nos parâmetros do TRI, apresentando baixa variabilidade dos fatores latentes.

Sumário

1. Introdução **7**

2. Fundamentação Teórica **11**

2.1. Teoria da Resposta ao Item 11

2.2. Sistemas de Recomendação 14

2.2.1. Algoritmos básicos 15

2.2.2. Algoritmos baseados em vizinhança 15

2.2.3. Algoritmos baseados em fatoração de matriz 16

3. Trabalho realizado **17**

3.1. Base de dados 17

3.2. Configuração dos algoritmos 20

3.3. Aplicação da TRI 20

4. Resultados **23**

4.1. Análise da habilidade dos respondentes 24

4.2. Análise dos parâmetros dos itens 27

5. Conclusão **34**

6. Referências Bibliográficas **36**

1. Introdução

O avanço da internet e a popularização da web trouxeram como consequência uma extensa quantidade de dados gerados por seus usuários. Esses dados, por sua vez, têm a potencialidade de serem transformados em informação, bem como em conhecimento. Todo esse novo ecossistema traz uma gama rica de escolhas, que ao invés de trazer benefícios para quem o possui, pode prejudicá-lo [1]. O poder da escolha remete ao sentido de liberdade e autonomia, mas em quantidade excessiva pode tornar o seu detentor confuso e paralisado [2].

Sistemas de recomendação são aplicações de filtragem de informação, e vêm se tornando uma forma eficiente para resolver esse problema de sobrecarga de conteúdo [3]. O objetivo desses sistemas, como o próprio nome sugere, é gerar recomendações ou sugestões para seu usuário, com base em seu histórico e preferências, tornando seu consumo mais acurado. A recomendação de filmes é uma aplicação amplamente utilizada na área, com vários trabalhos na academia e indústria, voltados a facilitar o acesso a conteúdos de interesse do usuário de uma forma mais inteligente [3]. O desafio lançado pela empresa Netflix [4], a qual buscava uma melhoria de ao menos 10% para seu sistema de recomendação, possibilitou que essa área alcançasse maior visibilidade e melhores algoritmos, se tornando bastante conhecida entre os desenvolvedores e trazendo novos entusiastas.

Atualmente existem inúmeros algoritmos com diversas estratégias para recomendação. A filtragem colaborativa (FC) é considerada a técnica mais popular e amplamente implementada nessa área, e a sua implementação mais simples e original recomenda ao usuário ativo os itens que outros usuários com gostos semelhantes gostaram no passado [8]. A similaridade entre dois usuários é baseado no histórico de avaliações dos itens em comum e pode ser calculado de várias formas, como por exemplo através do cálculo do cosseno.

Existem duas principais famílias de algoritmos de FC: baseado em vizinhança e fatoração matricial. Esses algoritmos utilizam um conjunto de dados como entrada,

o qual apresenta um padrão comum de atributos, com pequenas variações a depender da abordagem a ser empregada e a origem da base. Os dados pode ser obtidos atráves do feedback dos usuários, e o tipo mais conveniente é o feedback explícito de alta qualidade, no qual os usuários relatam diretamente seu interesse em produtos [8]. Essa categoria é exemplificada em serviços de streaming como a Netflix, onde o usuário após consumir algum item tem a liberdade de avaliá-lo com uma nota, geralmente de 1 a 5 estrelas, ou simplesmente com “gostei” ou “não gostei”, como é feito atualmente.

Cada sistema de recomendação consegue extrair diferentes informações, e consequentemente apresentar diferentes resultados para o usuário. Técnicas de FC podem ser avaliadas considerando duas perspectivas: baseada em posição (*ranking*) e avaliação (*rating*) [10]. Estas se referem ao tipo de retorno do algoritmo, a lista com os melhores itens para o usuário e a possível nota que o usuário dará para um determinado item, respectivamente. Para cada um existem diferentes formas de avaliar o desempenho do sistema, assim como qual o melhor contexto para utilizá-los.

É fato que a área de sistemas de recomendação dispõe uma gama rica e diversificada de algoritmos, cada um com suas particularidades de método, abordagem, tipo de entrada e resultados. Em muitos casos, é importante analisar os problemas específicos para os quais as melhores técnicas geralmente falham, enquanto outras mais simples podem ser bem-sucedidas [5]. Escolher uma técnica para melhor adequar ao cenário desejado pode demandar tempo e esforço, e para isso devem ser analisados e avaliados utilizando um conjunto de critérios e métricas.

A Teoria da Resposta ao Item (TRI) é um grupo de ferramentas de modelagem estatística utilizada na psicometria projetada para caracterizar de forma precisa os itens e os sujeitos [5], buscando representar a probabilidade que um indivíduo tem em acertar a resposta para um determinado item, em função dos parâmetros deste como sua dificuldade e discriminação, e também dos traços latentes do indivíduo, características que não podem ser observadas diretamente [20], como por exemplo a sua habilidade.

Os modelos de TRI são usados principalmente em testes educacionais e avaliação psicométrica, nos quais a capacidade dos examinados é medida usando

um teste com várias perguntas (ou itens) [5]. Eles resultam em atributos latentes dos respondentes como a habilidade, assim como a dificuldade e discriminação associados aos itens. Através dessa abordagem é possível observar o desempenho de cada sujeito e suas características quando respondem itens categorizados como fáceis e difíceis.

O estudo realizado em [5] traz a aplicação de um modelo TRI binário no contexto de aprendizagem de máquina, avaliando o desempenho de diferentes máquinas sob algumas bases de dados. Foram usados 128 classificadores a partir de 15 famílias de algoritmos, utilizando diferentes configurações de parâmetros, e dentre eles alguns classificadores artificiais (como por exemplo o péssimo e o ótimo, sempre falha e sempre tem sucesso, respectivamente, e randômicos) sendo considerados linhas de base a fim de comparação. As bases eram constituídas por problemas de classificação, com quantidades variadas de atributos e instâncias. Os resultados binários dos classificadores era obtidos sempre através do conjunto de teste e um modelo IRT, baseados em funções logísticas, era “aprendido” para cada instância da base ajustando a probabilidade de resposta correta dos classificadores de acordo com suas habilidades, sendo adotado o método da máxima verossimilhança para estimar os parâmetros dos modelos para todas as instâncias e a habilidades dos classificadores simultaneamente. Em [6] foi proposto um modelo TRI para respostas contínuas, com o objetivo em abordar as limitações conhecidas para esse tipo de modelo, propondo uma nova parametrização em relação à habilidade do respondente, e dificuldade e discriminação do item.

A adaptação de modelos TRI para problemas de classificação em aprendizagem de máquina abriu oportunidades para a aplicação em outras áreas de inteligência artificial, como por exemplo Sistemas de Recomendação. Este trabalho tem como objetivo utilizar a Teoria da Resposta ao Item como método avaliativo para algoritmos de recomendação no contexto de recomendação de filmes. Foram avaliados 10 algoritmos de filtragem colaborativa baseados em avaliação (*rating-based*), sob a base de dados por feedback explícito MovieLens com notas de 0 a 5. Para o modelo, os itens são os usuários da base com todo o seu histórico de avaliações, e os respondentes são os sistemas de recomendação. A resposta dos algoritmos é o erro associado à nota predita utilizando o RMSE, a qual serviu como

entrada para o modelo TRI, e este por fim estimando a habilidade latente dos sujeitos (algoritmos) e os parâmetros dos itens (usuários) como a dificuldade e a discriminação. Por tratar de resultados contínuos foi utilizado o modelo B³-IRT do estudo [6] que será discutido posteriormente.

Este trabalho está organizado da seguinte forma: a seção 2 apresenta a Teoria da Resposta ao Item, o modelo B³-IRT e os algoritmos de recomendação; a seção 3 explica a metodologia do trabalho realizado com o tratamento da base de dados, configuração dos algoritmos e a aplicação do TRI; a seção 4 traz a análise e discussão dos resultados quanto aos parâmetros de habilidade e dificuldade dos respondentes e itens, respectivamente; e por fim a seção 5 com a conclusão e considerações finais.

2. Fundamentação Teórica

Nesta seção, iremos introduzir a Teoria da Resposta ao Item e o modelo B³-IRT, e apresentar os sistemas de recomendação que foram avaliados nos experimentos com suas estratégias e propostas.

2.1. Teoria da Resposta ao Item

A Teoria da Resposta ao Item (TRI) é um grupo de ferramentas de modelagem estatística utilizada na psicometria projetada para caracterizar de forma precisa os itens e os sujeitos [5]. Ela se concentra nos itens, modelando as respostas dadas por indivíduos com habilidades diferentes a itens de diferentes dificuldades [6]. A definição de item depende do contexto no qual TRI é aplicado, podendo representar, por exemplo: questões objetivas de um teste, na área educacional; e instâncias de uma base de dados com determinados atributos e um rótulo (classe), na área de aprendizagem de máquina para problemas de classificação. Os respondentes, por sua vez, representam os indivíduos que irão fornecer suas respostas à determinados itens, como por exemplo, fazendo uma analogia aos exemplos de itens descritos anteriormente: estudantes e classificadores.

Modelos de TRI podem ser categorizados de acordo com a quantidade de parâmetros de seus itens: 1PL (parâmetro logístico) o qual consiste apenas no nível de dificuldade; 2PL além da dificuldade, o valor discriminativo; e 3PL, o mesmo que 2PL acrescentando a probabilidade de resposta correta do item ao acaso [21]. Para este trabalho será utilizado e discutido o modelo 2PL, não considerando o terceiro parâmetro presente em 3PL, mas a nível informativo, os respondentes com diferentes níveis de habilidade têm a mesma probabilidade de responder corretamente o item apenas adivinhando [22].

Modelos TRI 2PL estimam os parâmetros latentes dos itens, dificuldade e discriminação, e também traços latentes dos respondentes, como sua habilidade, com base nas respostas observadas em um teste, e têm sido comumente aplicados para avaliar o desempenho dos alunos nos exames [6]. Uma das aplicações mais

conhecidas é na área educacional, como por exemplo o ENEM [19], uma prova com questões objetivas aplicada aos alunos da rede de ensino pública e privada do Brasil ofertando vagas para as universidades de todo o país. Os sujeitos que tendem a responder corretamente os itens mais difíceis serão atribuídos a altos valores de habilidade, e os itens difíceis são aqueles respondidos corretamente apenas pelos respondentes mais proficientes [5].

Em modelos tradicionais de dois parâmetros a probabilidade de resposta dado a habilidade de um respondente é definida por uma função logística, descrita pela Curva Característica do Item (CCI). A intenção da curva é mostrar a relação da habilidade do respondente com a probabilidade de responder corretamente um item dado sua dificuldade e discriminação. Segundo [22], na Teoria da Resposta ao Item, a dificuldade de um item descreve onde o item se encontra na escala de habilidades dos respondentes, por exemplo, um item fácil se localiza entre os indivíduos pouco habilidosos, enquanto um item difícil entre aqueles com mais alta habilidade; portanto, a dificuldade modela o posicionamento da curva. Para o segundo parâmetro, a discriminação, descreve quão bem um item pode diferenciar entre respondentes com habilidades abaixo da localização do item e aqueles com habilidades acima desta, logo caracterizando a sua inclinação.

Retirada do artigo [22]

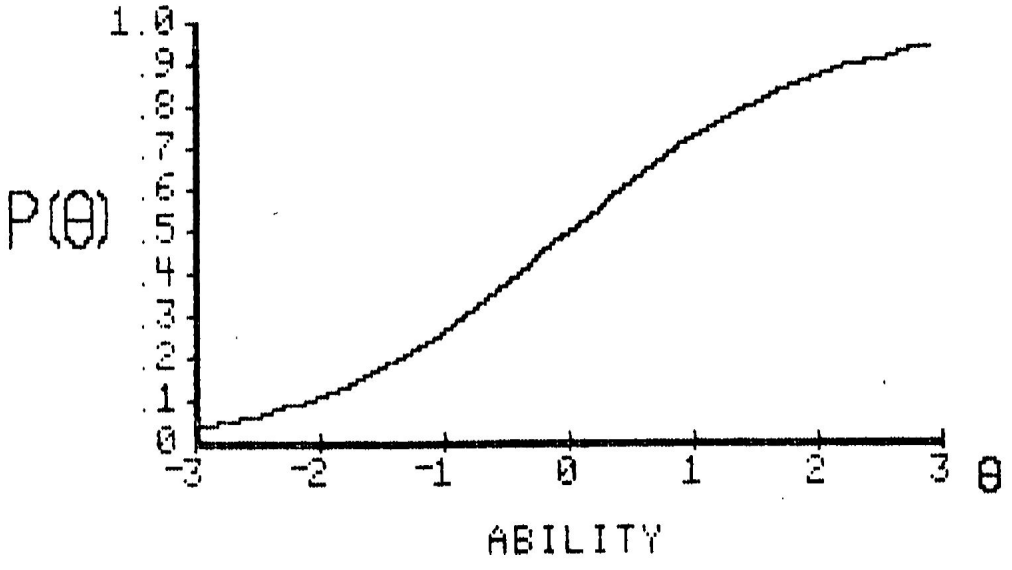

Figura 1: Exemplo de uma CCI.

A figura 1 exemplifica as características de uma CCI clássica. O eixo X representa a habilidade necessária do indivíduo para alcançar diferentes níveis de resposta (eixo Y), e essa relação é descrita em função dos parâmetros dos itens. Discriminações mais altas levam a ICCs mais íngremes, e itens mais difíceis precisam de respondentes mais hábeis para obter respostas mais altas [6].

A adaptação da TRI para outros contextos sugere novas perspectivas e possibilidades para diferentes aplicações além das previamente conhecidas. Muitos modelos TRI, como discutido no estudo [5], endereçam a problemas binários, que neste caso foi adaptado para analisar problemas de classificação em aprendizagem de máquina, em que os itens correspondiam às instâncias da base de dados e os classificadores aos respondentes, e suas respostas (certo ou errado à uma determinada classe) foram capturadas a partir da base de teste em experimentos de validação cruzada. Apesar do conhecimento obtido por este último trabalho, o estudo [6] propõe, posteriormente, um modelo IRT para respostas contínuas, o B³-IRT, abordando as limitações de modelos binários tradicionais quando há o interesse em probabilidade como retorno, como no próprio problema de classificação (porém agora multiclasse), como também as limitações conhecidas em modelos contínuos a nível de interpretabilidade da habilidade e dificuldade, e flexibilidade das CCIs que até então seguem a descrição de uma função logística.

Retirada do artigo [6]

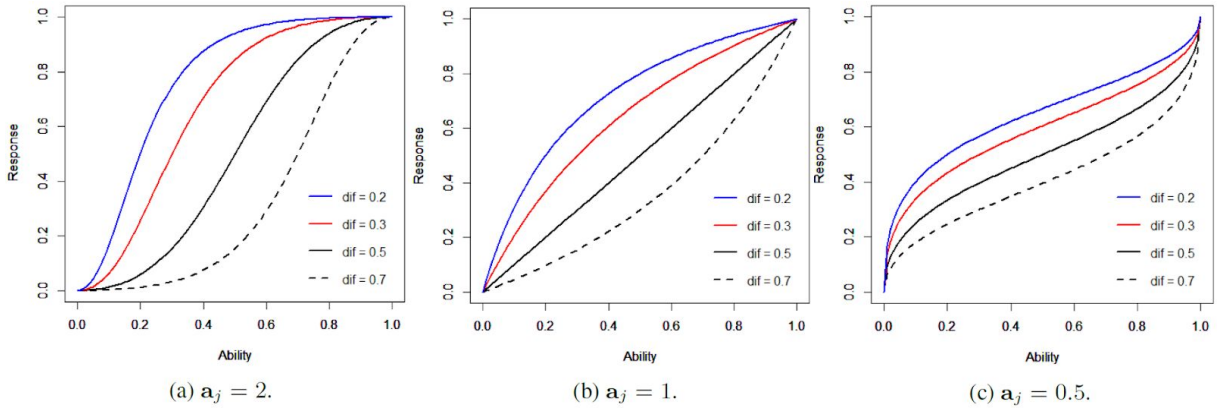

Figura 2: Exemplos de CCIs para o modelo B³-IRT para diferentes valores de dificuldade e discriminação.

O modelo B³-IRT compreende a categoria 2PL, definindo uma nova parametrização: a habilidade do respondente e a dificuldade do item em um intervalo entre 0 e 1, unificando a escala e facilitando a interpretação e avaliação; CCIs não limitadas a funções logísticas, tomando novas formas [6]. Itens mais discriminativos (figura 2.a) associados a baixa dificuldade (df = 0.2) resultam no efeito em que, com pouco aumento da habilidade a resposta esperada aumenta consideravelmente, ou seja, para a habilidade 0.2 a 0.4 temos um aumento de 0.5 para 0.9 na probabilidade de resposta correta, enquanto que com a mesma discriminação mas para itens difíceis (df = 0.7), apenas aqueles com maiores habilidades conseguirão melhores respostas.

2.2. Sistemas de Recomendação

Todos os algoritmos de recomendação utilizados neste estudo estão disponíveis na biblioteca Surprise [12], uma *engine* pública para sistemas de recomendação escrita em python, disponibilizando todo o ferramental necessário para o trabalho nessa área: algoritmos prontos para uso, plataforma para a criação de novos, modelagem e processamento de dados para adaptar ao formato aceito pela biblioteca, ferramentas para avaliação de desempenho por meio de algumas métricas conhecidas como MAE, MSE e RMSE, dentre outras *features*.

Foram selecionados 10 algoritmos previamente implementados pela Surprise, são eles: BaselineOnly, NormalPredictor, Co-Clustering, SlopeOne, KNNBasic e KNNWithMeans (UserBased e ItemBased), NMF e SVD. Todos baseados em *rating*, ou seja, a resposta é a predição da nota em que o usuário avaliaria um determinado filme. Os dados de entrada no processo de treinamento são no formato: ID do usuário, ID do filme e nota. Para a fase de teste é dado no mesmo formato, sendo retornado a nota real e a estimada. Os algoritmos podem ser divididos em 3 grupos de acordo com sua estratégia: básicos, baseados em vizinhança e baseados em fatoração matricial. Estes serão apresentados a seguir, baseados na documentação e referências da Surprise [12].

2.2.1. Algoritmos básicos

O algoritmo BaselineOnly prevê a estimativa “linha de base” para determinado usuário e item. A estimativa para uma nota desconhecida r_{ui}, nota do usuário u ao item i, é indicada por: $r_{ui} = \mu + b_u + b_i$ [13]. Os parâmetros b_u e b_i indicam os desvios padrões das médias observadas do usuário u e do item i, respectivamente, e μ a média geral de todos os filmes [13]. Dessa forma caso um filme seja avaliado acima da média geral, mas o usuário atribui notas abaixo desse limiar, a nota predita balanceará entre essas variáveis.

O algoritmo NormalPredictor prevê uma classificação aleatória com base na distribuição do conjunto de treinamento, gerada a partir da distribuição normal $\eta(\mu, \sigma)$ [12] onde μ é a média geral e σ é o desvio padrão de todos os filmes.

2.2.2. Algoritmos baseados em vizinhança

Os algoritmos KNNBasic e KNNWithMeans compartilham a tradicional estratégia base da filtragem colaborativa, em que a avaliação predita é em função da similaridade dos vizinhos mais próximos e suas respectivas notas para os itens em comum. A diferença do KNNWithMeans é que ele considera a média das notas de cada usuário pertencente à vizinhança. Foram utilizadas as abordagens *user-based* e *item-based* em cada algoritmo, e a função do cosseno para o cálculo da similaridade.

O algoritmo Slope One funciona com o princípio intuitivo de um "diferencial de popularidade" entre itens para os usuários, levando em conta as informações de outros usuários que classificaram o mesmo item e dos outros itens classificados pelo mesmo usuário [16]. O valor predito segue a seguinte função: $\widehat{r}_{ui} = \mu_u + \frac{1}{|R_i(u)|} \sum_{j \varepsilon R_i(u)} dev(i, j)$, onde μ_u é a média das notas do usuário u; $R_i(u)$ é o conjunto dos itens que foram avaliados por u e tem pelo menos um usuário em comum que avaliou i; e $dev(i, j)$ é a diferença média entre as notas dos itens i e j [13].

O Co-Clustering pode ser visto como um método para agrupar dois tipos de entidades simultaneamente, com base na similaridade de suas interações em par a par [17]. O nota predita segue a seguinte função: $\widehat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i})$ onde $\overline{C_u}$, $\overline{C_i}$ e $\overline{C_{ui}}$ são as médias das notas dos *clusters* do usuário u, do item i e do co-clustering de ambos, respectivamente; e μ_u e μ_i são as médias das notas de u e i nessa ordem [12].

2.2.3. Algoritmos baseados em fatoração de matriz

O algoritmo SVD foi popularizado por Simon Funk [18] quando a Netflix lançou seu desafio de recomendação [4]. Similar ao Non-negative Matrix Factorization (NMF), essas abordagens mapeiam usuários e itens para um espaço de fator latente de modo que as interações usuário-item sejam modeladas como produtos internos nesse espaço [14]. A diferença básica entre os dois algoritmos é que os fatores dos usuários e itens são positivos e negativos no SVD e apenas positivo no NMF.

3. Trabalho realizado

TRI é amplamente conhecida na área educacional como método avaliativo do desempenho de estudantes, e em trabalhos recentes foi adaptado para a aplicação no contexto de aprendizado de máquina, em modelos binários, para análise da performances dos diferentes algoritmos de classificação [5], como também para modelos contínuos [6].

Este trabalho tem como proposta aplicar o modelo B³-IRT [6] para avaliar diferentes sistemas de recomendação, em contraste às métricas comumente utilizadas como a RMSE, buscando entender os cenários em que algumas estratégias utilizadas por esses algoritmos conseguem obter melhores resultados em comparação às outras. Para a TRI aplicada ao contexto deste trabalho, os itens são os usuários da base e os respondentes são os algoritmos de recomendação.

Nesta seção iremos detalhar sobre o trabalho desenvolvido, abordando aspectos a respeito dos dados utilizados e seu pré-processamento na subseção 3.1, a configuração dos parâmetros dos algoritmos utilizados na subseção 3.2 e por fim a aplicação do modelo de Teoria da Resposta ao Item, o B³-IRT, e a apresentação do fluxo do trabalho realizado na subseção 3.3.

3.1. Base de dados

Os dados utilizados pertencem ao *GroupLens*, um grupo de pesquisa do Departamento de Ciência da Computação e Engenharia da Universidade de Minnesota. Este grupo coleta e torna disponível bases de dados com as notas de usuários do serviço web de recomendação de filmes *MovieLens* [7]. Este conjunto de dados contém 11 níveis de classificação, entre 0 a 5 estrelas. Esses dados foram criados por 610 usuários, dispondo de 100.836 classificações em 9.742 filmes [7].

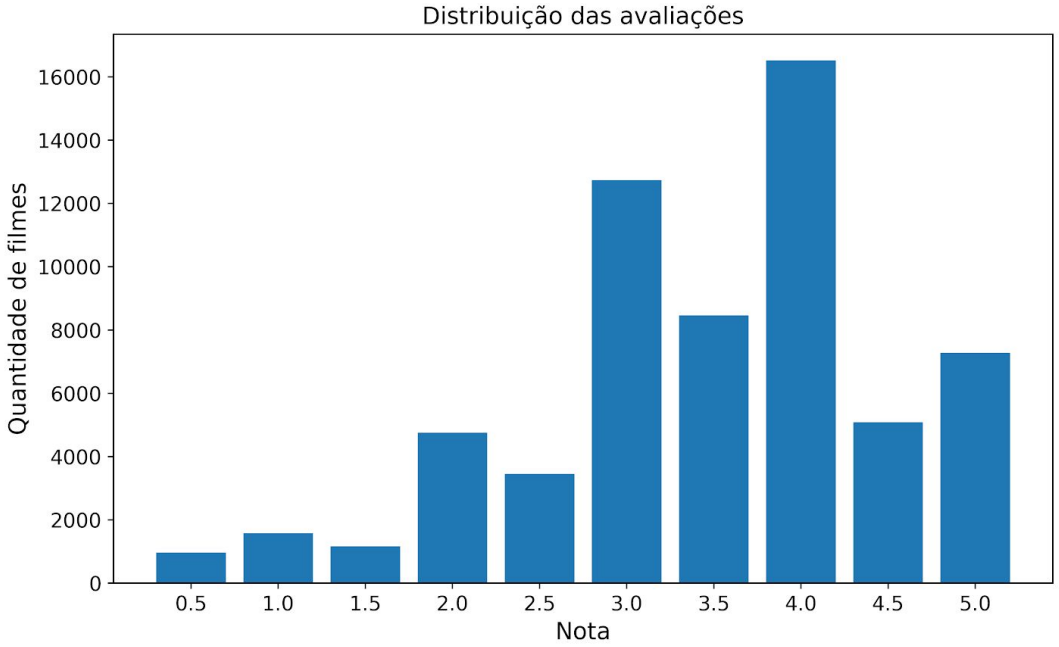

Figura 3: Distribuição das notas de 0 a 5 dadas pelos usuários aos filmes.

Para este estudo a base de dados passou por uma etapa de filtragem e pré-processamento. Primeiramente foram considerados apenas os usuários que tenham avaliados ao menos 30 filmes. Como os dados apresentam uma distribuição exponencial para quantidade de filmes avaliados por usuários, sua densidade se concentra entre 30 a 100 filmes, com média total de 206 filmes. A base foi dividida em 3 seções, cada uma com quantidades próximas de usuários, intervaladas em quantidade de itens avaliados, para que houvesse variabilidade no conjunto final. Os intervalos considerados foram: 30 a 60 filmes; 61 a 154; e 155 ao máximo disponível.

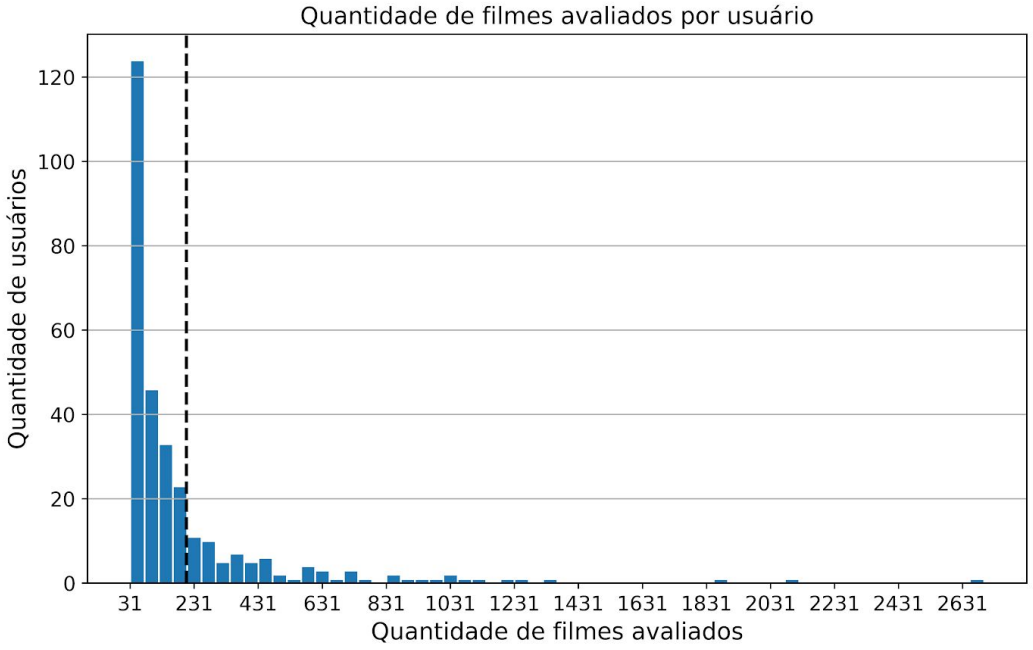

Figura 4: Distribuição da quantidade de avaliações por usuário.

Após essa etapa, 100 usuários de cada intervalo foram selecionados de forma aleatória, formando assim o conjunto de dados que será discutido e analisado no decorrer deste trabalho. A tabela 1 mostra a comparação entre a base original e a pós-processada, utilizada no estudo.

Base	**# Usuários**	**# Filmes**	**# Notas**
Original	610	9.742	100.836
Pós-processada	300	8.181	62.001

Tabela 1: Comparação entre as bases de dados original e pós-processada

Especificado qual o conjunto de dados que seria trabalhado, foi realizado a divisão em dois sub-conjuntos, treinamento e teste, correspondendo 70% e 30% respectivamente. Essa repartição foi feita para os filmes avaliados de cada um dos 300 usuários, ou seja, resultando em duas bases com os mesmos usuários, divergindo apenas nos filmes e suas devidas notas.

A base de dados dispõe das seguinte informações: ID do usuário, ID do filme, nota e o horário em que a nota foi registrada. Para simplificação, apenas os três primeiros foram utilizados no decorrer do estudo.

3.2. Configuração dos algoritmos

Como mencionado anteriormente, foram utilizados dez algoritmos de recomendação de diferentes abordagens, agrupados em: básicos; baseados em vizinhança; e baseados em fatoração matricial. Todas as implementações foram obtidas através da biblioteca Surprise [12], uma *engine* pública para sistemas de recomendação, escrita em python, disponibilizando algoritmos, processamento de dados e ferramentas para avaliar a performance por meio de algumas métricas conhecidas como MAE, MSE e RMSE.

Os algoritmos requerem diferentes tipos de configurações em seus parâmetros devido às suas propostas, entretanto para o presente trabalho foi preservado a maioria das definições padrões previamente estabelecidas pela biblioteca. Para os algoritmos SVD, NMF e Co-Clustering foram estabelecidos o mesmo valor (42) para a semente randômica, utilizada na fase de inicialização. Para os algoritmos que utilizam o método de vizinhança, como o KNNBasic e o KNNWithMeans, apenas a medida de similaridade foi definida, sendo utilizada a do cosseno. O estimador do BaselineOnly foi configurado para usar o método Gradiente Descendente Estocástico (SGD). Por fim, para o SlopeOne e o NormalPredictor não tiveram nenhuma mudança em seus parâmetros.

3.3. Aplicação da TRI

Como mencionado anteriormente TRI é amplamente utilizado na área da psicometria como ferramenta para avaliar a capacidade latente humana por meio de testes [6]. Modelos tradicionais utilizam conjuntos de respostas binárias (correto ou errado) para o seu funcionamento, limitando os possíveis cenários os quais poderiam utilizar dessa ferramenta para avaliação de performance. Tendo isso em vista foi utilizado

neste trabalho o B³-IRT, o qual busca abordar esta e outras limitações conhecidas, propondo um modelo TRI para respostas contínuas, aplicando o mesmo às probabilidades preditas e utilizando uma nova forma de parametrização [6]. Para o presente contexto foram definidos como itens e respondentes, os usuários da base e os algoritmos de recomendação, respectivamente.

A partir da definição dos itens para o modelo TRI o processo de treinamento e teste dos algoritmos deveriam se ajustar a esse cenário. Dessa forma todos os sistemas de recomendação foram treinados da seguinte forma: todos os subconjuntos de treinamento (e.g. 70% das avaliações de cada usuário) foram unidos, transformando em um único conjunto sendo aplicado nesta etapa inicial, utilizando as configurações dos parâmetros previamente apresentadas. Na etapa seguinte os sistemas de recomendação foram avaliados utilizando a métrica da raiz do erro médio quadrático (RMSE). Os subconjuntos de teste eram preditos um a um para cada usuário, dessa maneira teríamos o resultado dos respondentes para todos os itens da base.

Após todos os resultados serem obtidos, foi construída a Matriz Resposta, uma matriz I x R onde I é a quantidade de itens avaliados (linhas) e R o número de respondentes (colunas). Este é o modelo de entrada esperado pelo B³-IRT, porém os valores devem estar no intervalo aberto entre 0 e 1, sendo interpretado na escala de pior à melhor resposta, respectivamente. Para a normalização dos resultados foi utilizado a equação 1, onde ε_{ir} é a raiz do erro médio quadrático do respondente r para o item i.

$$\frac{1}{1+\varepsilon_{ir}} \qquad (1)$$

O modelo B³-IRT utilizado tem como entrada a Matriz Resposta e alguns parâmetros de configuração: nome da base de dados, tamanho da base, uma fração de ruído e um valor para inicializar um gerador randômico (semente). Os valores utilizados foram: "*recommendation*", 300, não aplicado, e 1, respectivamente. O nome é apenas utilizado para identificação no momento da geração dos resultados; o tamanho da base se refere à quantidade de linhas da matriz (itens/usuários); para

este trabalho não foi adicionado nenhum tipo de ruído aos dados; e o valor da semente randômica foi escolhida sem nenhum critério, de forma aleatória.

A figura 5 apresenta o fluxograma da aplicação do TRI discutido nesta seção: 1) filtragem e pré-processamento da base de dados original resultando na pós-processada com 300 usuários; 2) divisão entre dados de treinamento e teste, 70% e 30%, respectivamente, treinamento e avaliação dos algoritmos para cada usuário, utilizando RMSE normalizado para gerar a Matriz Resposta; e 3) aplicação do B³-IRT resultando duas saídas: habilidade dos respondentes e os parâmetros dos itens.

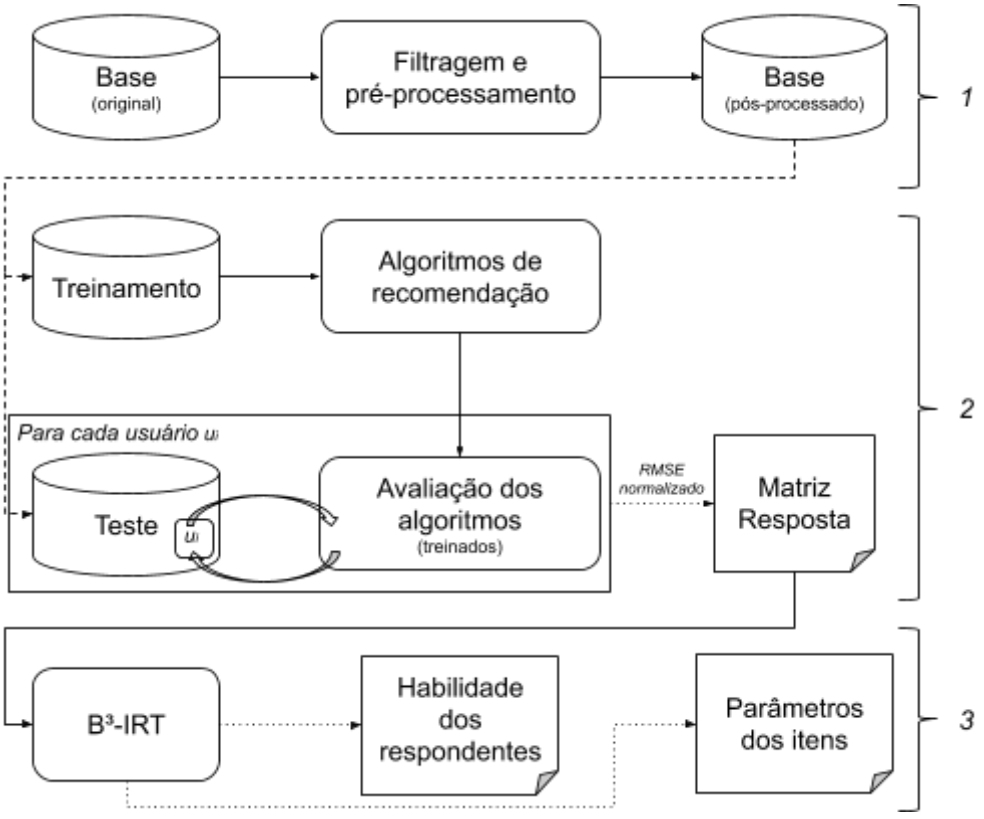

Figura 5: Fluxograma da aplicação do TRI

Todas as etapas da infraestrutura presente na figura 5, incluindo scripts de processamento dos dados e resultados, implementação dos algoritmos de recomendação e do modelo B³-IRT foram desenvolvidos utilizando primariamente Python 3.

4. Resultados

Nesta seção são apresentados os resultados obtidos da predição dos sistemas de recomendação, e da aplicação do modelo B³-IRT no mesmo contexto. Inicialmente os algoritmos foram avaliados através da métrica RMSE para cada usuário da base, como explicado na seção anterior. A partir desse ponto, esses valores foram normalizados e utilizados como entrada para o modelo TRI previamente apresentado, gerando os resultados de habilidade e dificuldade dos respondentes e itens. Para este trabalho cada respondente é um algoritmo de recomendação e os itens são os usuários da base de dados.

A tabela 2 abaixo apresenta os valores médios dos resultados para cada algoritmo, ordenada pelo valor de RMSE. Destaque para os algoritmos BaselineOnly e SVD com os melhores resultados dentre os concorrentes, os KNNWithMeans performaram melhor em relação aos KNNs simples e o NormalPredictor com o pior desempenho. Esses resultados iniciais não consideram o modelo TRI, apenas uma análise simples através de uma métrica tradicional.

Algoritmo	**RMSE**
BaselineOnly	0,8439
SVD	0,85
SlopeOne	0,8734
KNNWithMeans_ItemBased	0,8736
KNNWithMeans_UserBased	0,8760
Co-Clustering	0,9067
NMF	0,9092
KNNBasic_ItemBased	0,9145
KNNBasic_UserBased	0,9452
NormalPredictor	1,4076

Tabela 2: RMSE médio dos algoritmos de recomendação

Como dito anteriormente os resultados dos algoritmos (RMSE) foram normalizados utilizando a equação (1) dentro do intervalo aberto entre 0 e 1, para se adequar a entrada esperada pelo modelo B³-IRT, do pior para o melhor desempenho respectivamente. A seguir, a seção 4.1 apresenta a discussão da habilidade dos algoritmos (respondentes), e a seção 4.2 sobre os parâmetros de dificuldade e discriminação dos usuários (itens), e sempre que houver menção sobre “resposta” ou “resposta média” dos sistemas de recomendação será referente ao valor normalizado.

4.1. Análise da habilidade dos respondentes

A figura 6 mostra a relação entre a resposta média e a habilidade de cada algoritmo. Diferentemente como evidenciado pelo estudo [6] em que há forte correlação entre esses atributos, aqui temos ainda que um efeito positivo com coeficiente 0.3212, não é estatisticamente significativo (p-value 0.3677) [11]. Podemos observar a característica descrita na tabela 2 do desempenho geral semelhante entre os sistemas de recomendação sendo refletida no eixo Y deste gráfico, porém a habilidade avaliada pelo modelo B³-IRT não mantém o mesmo ranking por desempenho. Devido ao resultado inferior do algoritmo NormalPredictor em todos os aspectos descrito na tabela 3 e para simplificar a análise dos resultados, este será desconsiderado nas discussões a seguir.

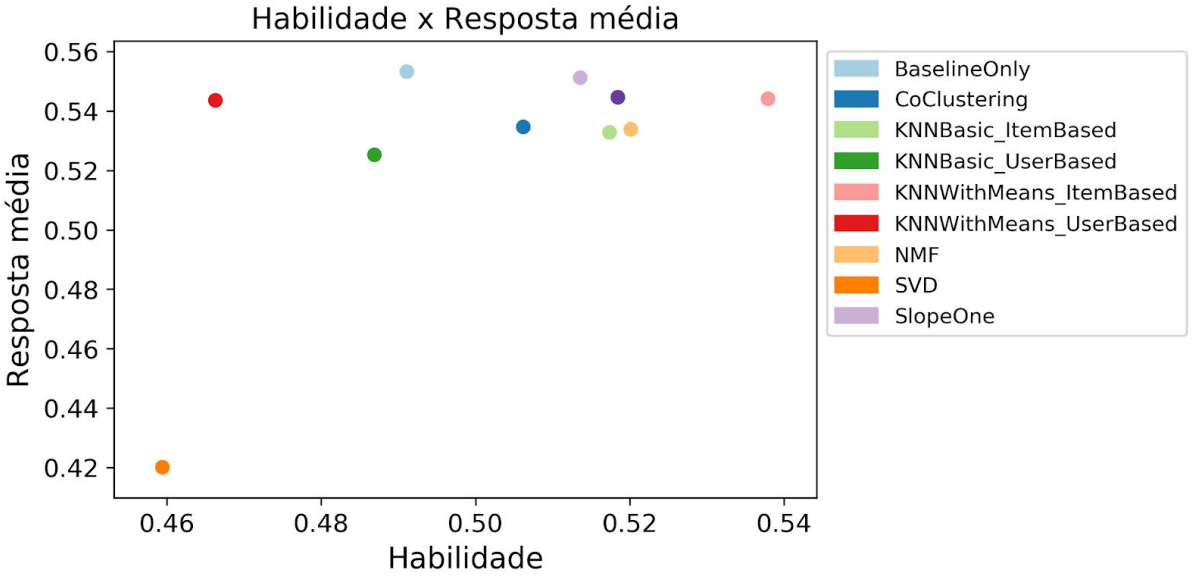

Figura 6: Resposta média vs. habilidade dos algoritmos de recomendação

A tabela 3 sumariza as informações sobre RMSE, habilidade e resposta média para cada algoritmo. Assim podemos destacar, a respeito da habilidade, o KNNWithMeans_ItemBased com o melhor rendimento seguido do NMF, um grupo formado pelo SlopeOne, KNNBasic_ItemBased e SVD, e como esperado o NormalPredictor com a pior performance. Dessa forma os KNNs (WithMeans e Basic) que utilizam o método de similaridade baseado no item, nesse caso os filmes, obtiveram melhor desempenho. A habilidade de um respondente não é medida em termos do número de respostas corretas, mas é estimada com base em suas respostas a itens discriminantes com diferentes níveis de dificuldade [5].

Algoritmo	**Habilidade**	**Resposta Média**
KNNWithMeans_ItemBased	**0.5379**	0.5440
NMF	0.5201	0.5337
SlopeOne	0.5184	0.5445
KNNBasic_ItemBased	0.5173	0.5327
SVD	0.5135	0.5511
CoClustering	0.5062	0.5346
BaselineOnly	0.4911	**0.5532**
KNNBasic_UserBased	0.4869	0.5252
KNNWithMeans_UserBased	0.4663	0.5435
NormalPredictor	0.4594	0.42

Tabela 3:Habilidade (B^3-IRT) e resposta média dos algoritmos de recomendação

Podemos ver que a habilidade se comporta de maneira diferente das outras métricas, uma vez que não é apenas estimada usando agregados de probabilidades previstas, mas também pelas dificuldades e discriminações dos itens correspondentes [5]. Assim notamos que o baixo erro associado ao BaselineOnly e uma das maiores médias de resposta dentre os demais (diferença de 0.009 para o

KNNWithMeans_ItemBased) não lhe garantiu a melhor habilidade no contexto dos dados utilizados.

A figura 7 mostra um comparativo entre os algoritmos KNNWithMeans_UserBased (A) e BaselineOnly (B), destacando o nível de dificuldade dos itens (parâmetro que será discutido na próxima subseção). A habilidade de A foi a menor do grupo (exceto por NormalPredictor), enquanto B obteve a maior média de resposta, no gráfico podemos observar que apesar da pouca dispersão dos pontos, estes estão localizados ligeiramente acima da reta de referência (linha pontilhada) assim como alguns pontos mais escuros (difíceis). Embora o fator latente da habilidade é definido considerando outros aspectos, percebe-se uma distribuição favorável ao algoritmo B.

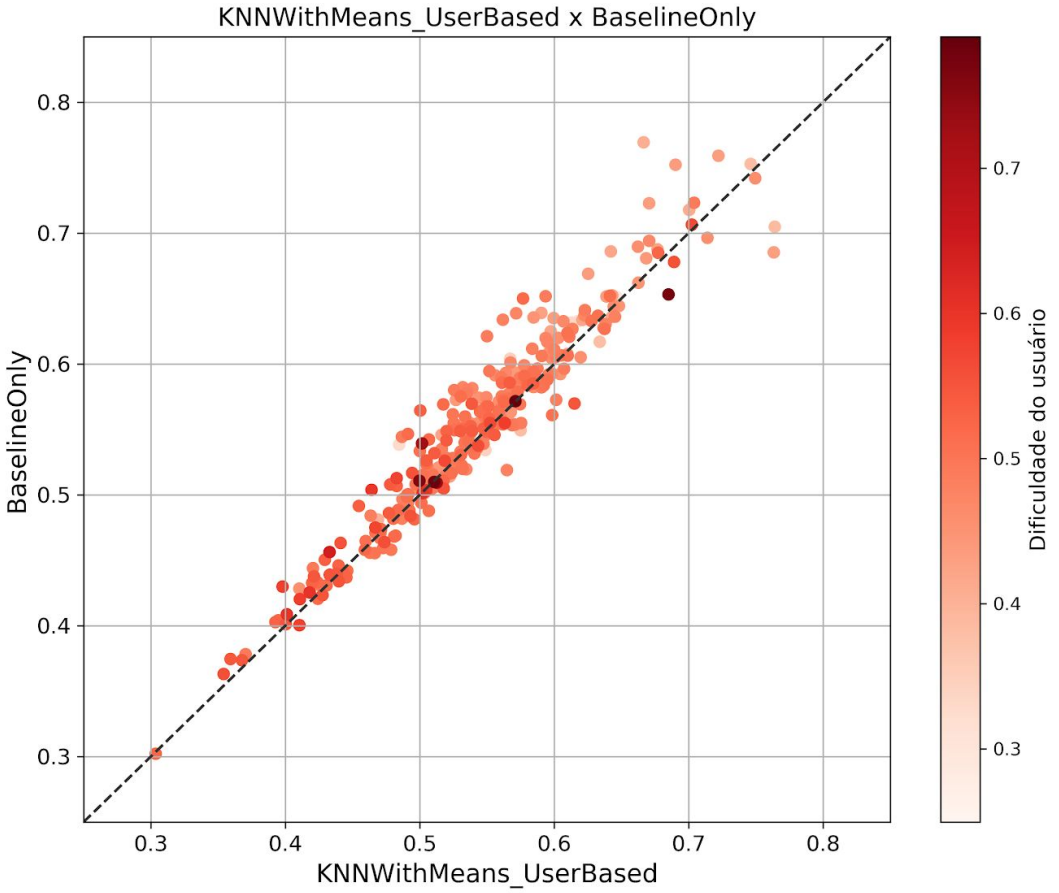

Figura 7: Comparação entre respostas médias do KNNWithMeans_UserBased e BaselineOnly, por nível de dificuldade dos itens

Fazendo a mesma comparação entre KNNWithMeans_UserBased (A) e KNNWithMeans_ItemBased (B) temos a figura 8. Ambos algoritmos obtiveram

valores RMSE e resposta média próximos, porém B é superior em habilidade. Esse cenário é menos disperso que o anterior, tornando a visualização mais complexa, porém pontos mais escuros são observados um pouco acima da diagonal.

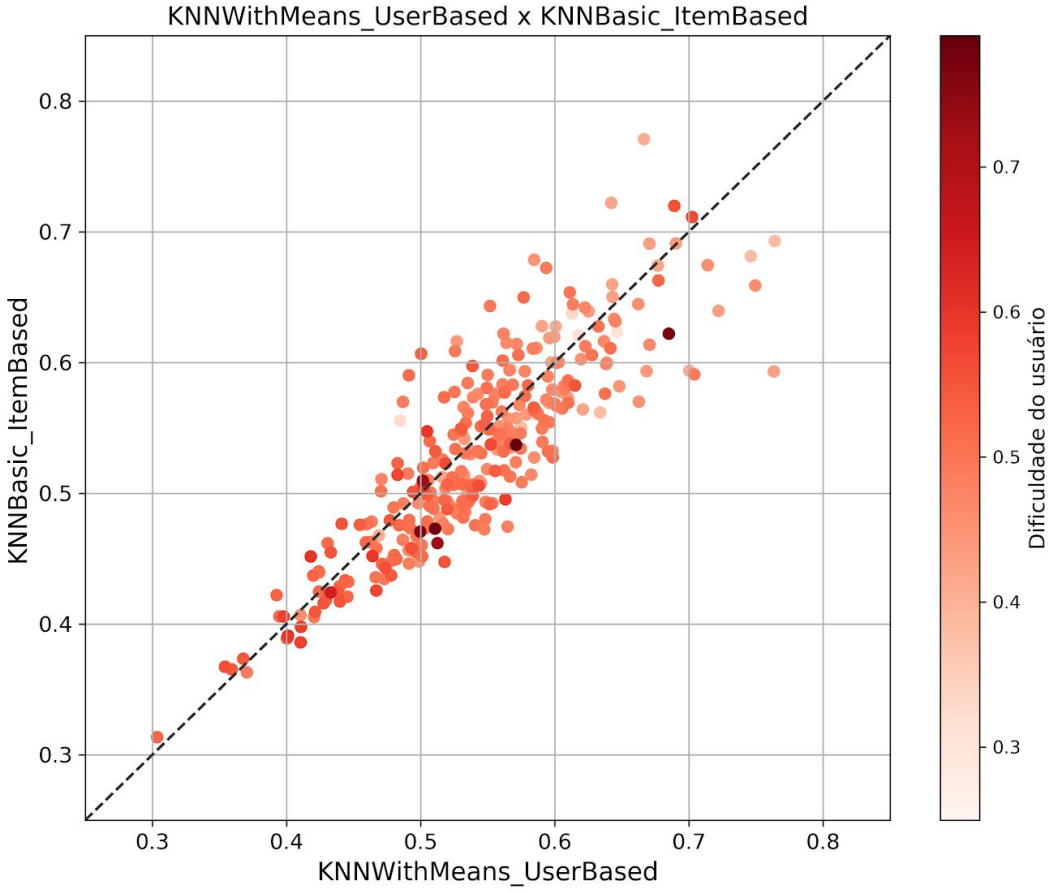

Figura 8: Comparação entre respostas médias do KNNWithMeans_UserBased e KNNWithMeans_ItemBased, por nível de dificuldade dos itens

4.2. Análise dos parâmetros dos itens

As figuras 9 e 10 mostram as distribuições da dificuldade e discriminação dos 300 itens (usuários) avaliados na base de dados MovieLens. Ambas apresentam o comportamento de uma distribuição normal: 44% aproximadamente dos itens se agrupam no nível “médio” de dificuldade em torno de 0,5 (parâmetro entre 0 e 1), enquanto cerca de 3% com ao menos 0.6 e outros 3% com no máximo 0,4; para esse contexto dos dados 33% aproximadamente se encontram no nível “médio” de discriminação em torno de 0,3 e apenas 3,6% com valores negativos. Não existe intervalo esperado para este último parâmetro, por isso para esse contexto 0,3 pode

ser considerado um nível médio, entretanto em comparação a outros estudos como em [5 e 6] esses itens não possuem um alto fator distintivo. O parâmetro de discriminação é uma medida da capacidade de um item diferenciar entre indivíduos (respondentes) [5].

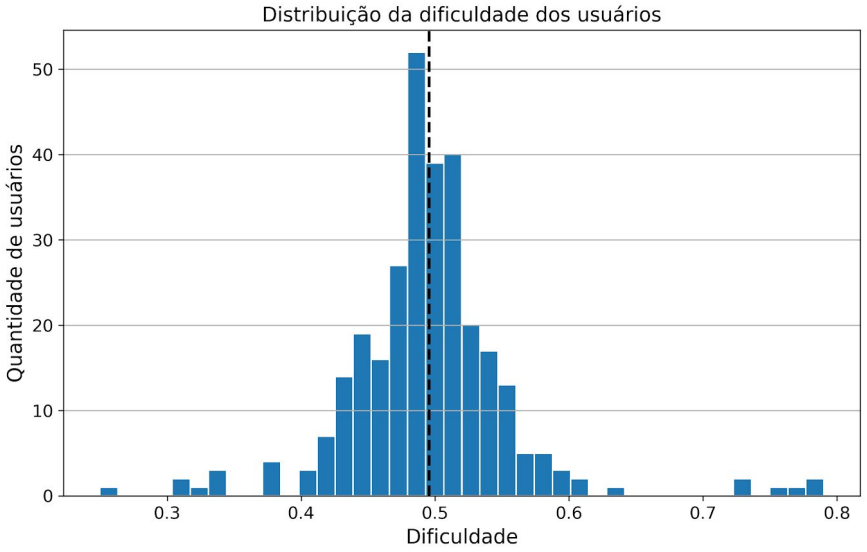

Figura 9: Distribuição da dificuldade dos itens (usuários). Média: 0.495, Desvio padrão: 0.062.

A TRI mostra um comportamento duplo na maneira como a habilidade do respondente e a dificuldade do item são estimadas ao mesmo tempo, dependendo dos outros candidatos e itens [5]. As performances aproximadas dos algoritmos refletiram no nível de dificuldade dos usuários, e vice versa, visto que não houve uma quantidade significativa de casos em que itens tiveram ótimas respostas e respondentes se destacaram em desempenho em detrimento dos demais.

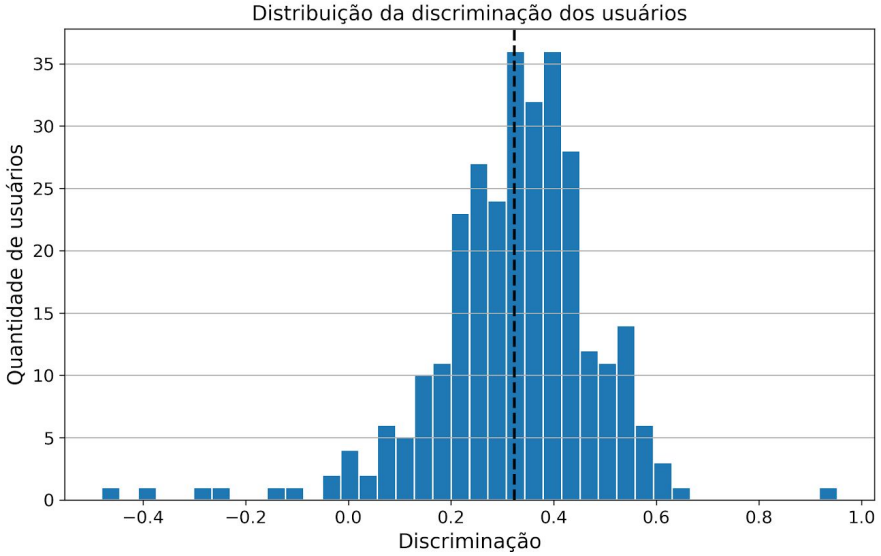

Figura 10: Distribuição da discriminação dos itens (usuários). Média: 0.323, Desvio padrão: 0.159.

A figura 11 descreve o comportamento das respostas dos respondentes em função da dificuldade dos itens. Dividimos os usuários em 30 grupos de mesmo tamanho, ordenados por dificuldade, e para cada grupo plotamos no eixo X a média deste parâmetro e no eixo Y a média de resposta de cada algoritmo. Nesse gráfico podemos observar, o decaimento de forma conjunta do resultado dos sistemas de recomendação, ainda que há pequenas variações, de 0,62 para 0,47 aproximadamente. Apesar do agrupamento em pequenos *clusters*, alguns algoritmos se destacaram em picos como: BaselineOnly, SVD e KNNWithMeans (UserBased e ItemBased), e vales: KNNBasic (UserBased e ItemBased) e NMF.

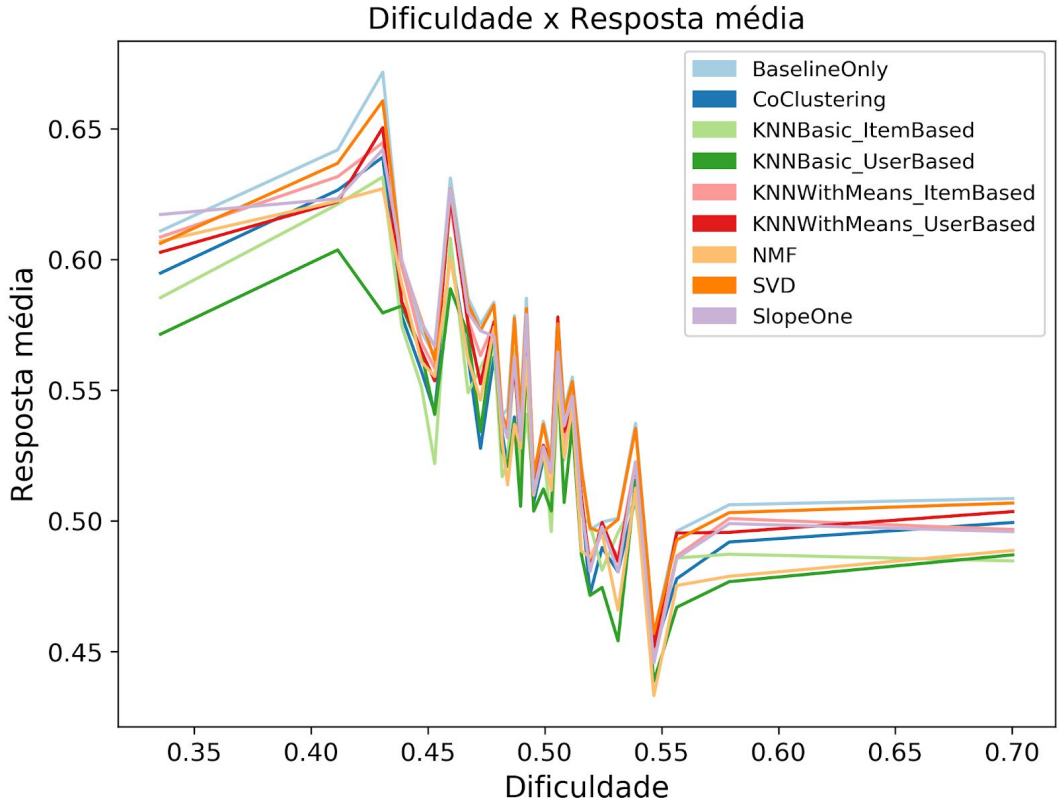

Figura 11: Resposta média à medida que aumenta a dificuldade dos itens (exceto NormalPredictor)

Contrastando os parâmetros de dificuldade e discriminação, embora este último apresentando baixos valores, temos a distribuição dos itens em relação a ambos mostrada na figura 12. É possível observar que, além do já esperado aglomerado de itens localizado no centro (os valores médios), há pequenos grupos nos 4 extremos: (1) dificuldade média e discriminação baixa; (2) dificuldade alta e discriminação média; (3) dificuldade baixa e discriminação média; e (4) dificuldade média e discriminação alta.

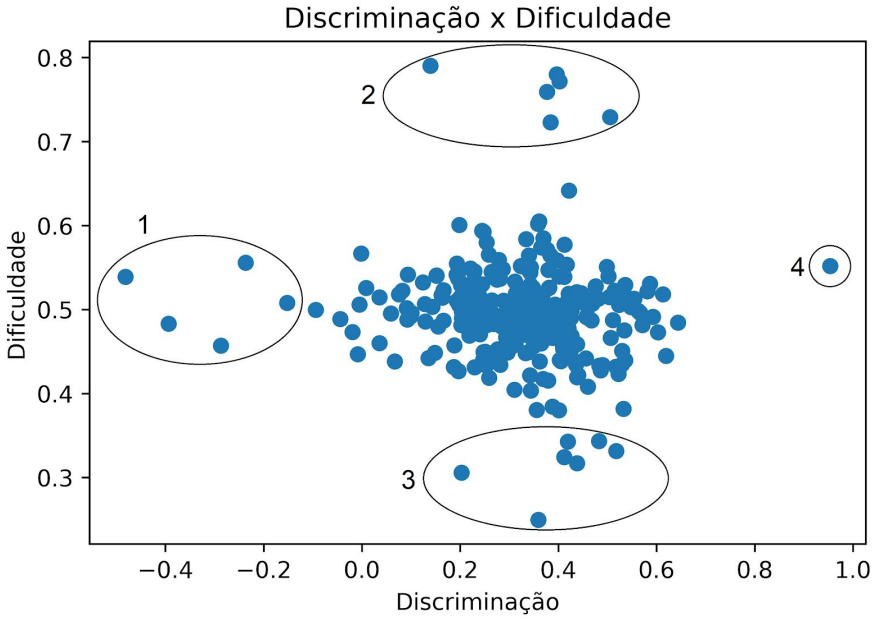

Figura 12: Comparação entre dificuldade e discriminação dos itens, formando 4 grupos..

Após a identificação dos grupos, foi selecionado um indivíduo de cada um, e realizado análises mais específicas. Para isso utilizamos a Curva Característica do Item (CCI) de cada usuário, uma função logística que retorna a probabilidade de uma resposta correta para o item com base na habilidade do respondente [6]. Ela ajuda a visualizar os três parâmetros da TRI: habilidade do algoritmo e sua resposta ao item; a dificuldade que reflete como o parâmetro de localização da curva; e a discriminação que ajusta a inclinação.

A figura 13 abaixo mostra um exemplo para um usuário de cada grupo. Os pontos coloridos são os sistemas de recomendação utilizados neste estudo, e seu posicionamento reflete o comportamento mostrado pela figura 6. Na figura 13a temos um exemplo do grupo 1, o único grupo com o sentido da curva oposta em relação aos demais devido à sua discriminação negativa, e que segundo a sua leitura os respondentes com menos habilidades alcançam maiores respostas em relação aos sujeitos mais hábeis, o que nesse caso o KNNWithMeans_UserBased obteve a maior resposta (0.6), apesar da pequena diferença em relação ao segundo melhor para esse item, e com exceção ao NormalPredictor pela sua pior performance em todos os casos.

As figuras 13b e 13c apresentam características em comum como a inclinação da reta (valores próximos de discriminação) e o agrupamento dos

respondentes. Por se localizarem nos extremos em nível de dificuldade, as curvas apresentam diferentes alturas, com pouco aumento da habilidade a resposta esperada em 9b inicia em 0.2 até 0.4 para uma habilidade de 0.5, enquanto em 9c, para o mesmo ritmo de crescimento, inicia em 0.4 até 0.6 para uma habilidade de 0.5.

A figura 13d representa o item com maior discriminação da base, por seu valor ser próximo a 1 a característica da curva é se aproximar de uma reta, tornado a relação entre resposta esperada e habilidade linear. Este último é o usuário com maior quantidade de filmes avaliados dentre os 4, a menor média e maior desvio padrão, apresentado uma distribuição normal sobre suas notas. A tabela 4 sumariza os parâmetros das CCIs e as informações sobre quantidade de filmes avaliados, e média e desvio padrão das notas para cada usuário presentes na figura 13.

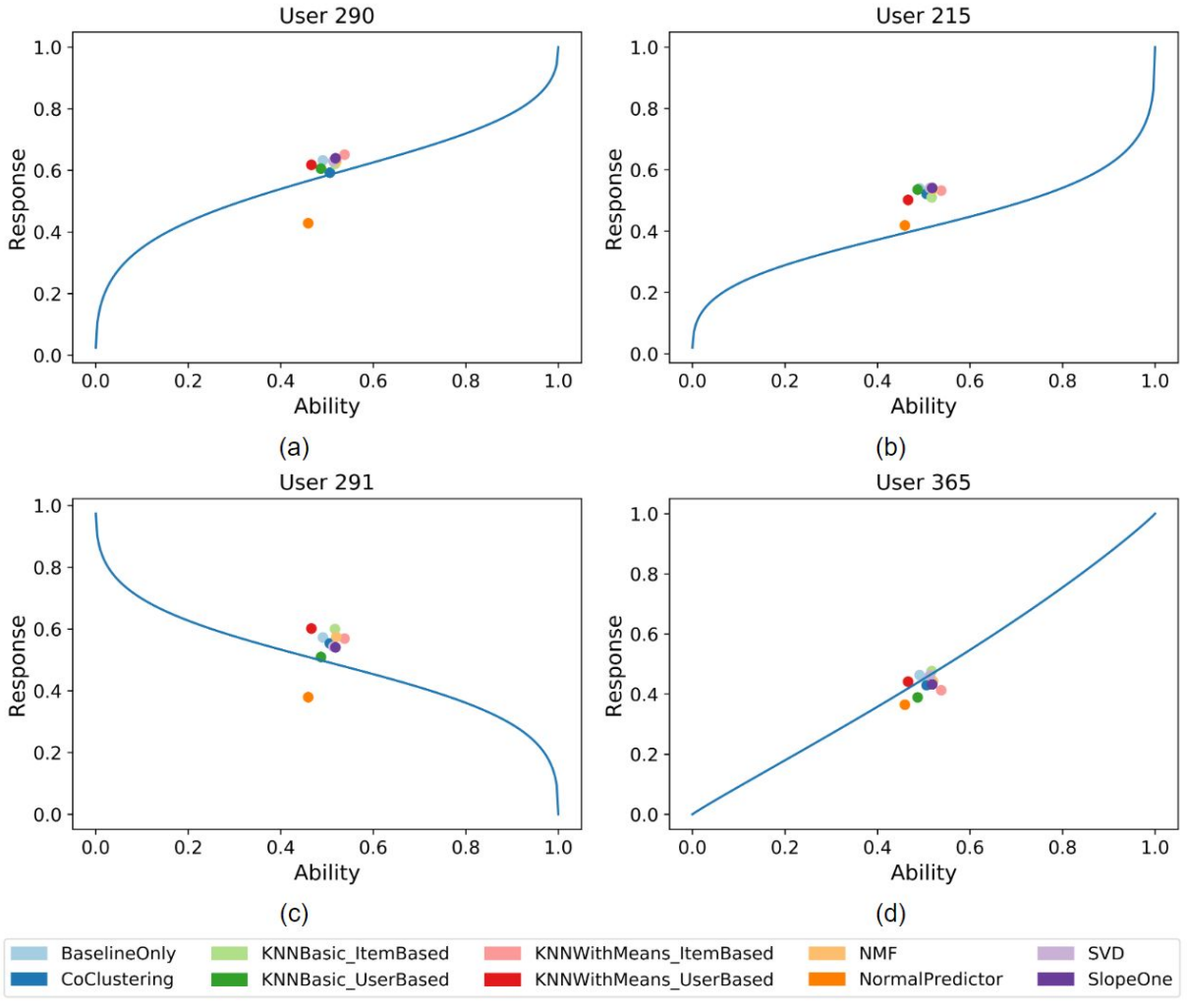

Figura 13: Exemplos de CCI para os 4 grupos: (a) grupo 1, (b) grupo 2, (c) grupo 3 e (d) grupo 4. Círculos coloridos são as respostas dos algoritmos.

Usuário	# Filmes	Nota média	Desvio Padrão (Nota)	Dificuldade	Discriminação
291	31	4,25	0,89	0,48	-0,39
215	98	3,9	0,86	0,72	0,38
290	297	4,14	0,68	0,31	0,43
365	277	2,75	1,07	0,55	0,95

Tabela 4: Quantidade de filmes avaliados, média e desvio padrão das notas por usuário, e parâmetros das CCIs da figura 13.

5. Conclusão

Neste trabalho foi realizado um estudo sobre a utilização da Teoria da Resposta ao Item como forma avaliativa para sistemas de recomendação em contraste às tradicionais métricas como o RMSE. Foram selecionados 10 algoritmos de recomendação (*rating-based*) por filtragem colaborativa, divididos em 3 grupos: básicos, baseados em vizinhança e baseados em fatoração matricial. A base de dados por feedback explícito da MovieLens continha notas de 0 a 5, da qual foram escolhidos 300 usuários aleatoriamente, e para cada um 70% dos filmes separados para treinamento e os outros 30% para teste.

Para a TRI foi utilizada o modelo B³-IRT [6], definindo os algoritmos como sendo os respondentes e os usuários como sendo os itens. Após realizada as recomendações, o RMSE normalizado serviu como entrada para o modelo, o qual resultou nos fatores latentes da habilidade dos respondentes, e dificuldade e discriminação dos itens. Os experimentos mostraram resultados diferentes dos esperados pelas métricas tradicionais. BaselineOnly e SVD alcançaram o menor RMSE (~0,85), porém os algoritmos KNNWithMeans_ItemBased e NMF obtiveram as maiores habilidades (~0,53), em um intervalo aberto entre 0 e 1, e 44% dos usuário foram classificados com nível “médio” de dificuldade (~0,5), no mesmo intervalo.

A avaliação dos sistemas de recomendação através da TRI apresentou uma análise sobre o desempenho dos mesmos em uma nova perspectiva, itens com diferentes níveis de dificuldade e discriminação. Entretanto, o desempenho similar desses algoritmos influenciaram nos parâmetros do TRI, apresentando baixa variabilidade dos fatores latentes, uma vez que esse método infere simultâneamente. Parte significativa dos usuários foram agrupados entre 0,4 e 0,6 no parâmetro de dificuldade, um valor médio para a escala, e baixíssimos índices a nível de discriminação, em comparação a outros estudos [5 e 6], dificultando a distinção entre os algoritmos e a interpretação dos resultados e seus comportamentos.

Devido esse efeito nos resultados, abre-se passagem para possíveis trabalhos futuros, a fim de realizar diferentes experimentos nesse cenário de recomendação de filmes: aumentar a quantidade e diversidade dos dados, buscando novas fontes além da MovieLens; novas técnicas de recomendação, como por exemplo baseada em conteúdo ou híbrido; trocar a perspectiva de avaliação para *ranking-based* ou utilizá-la juntamente com *rating-based*.

6. Referências Bibliográficas

[1] RICCI, Francesco. Recommender Systems: Models and Techniques. Encyclopedia Of Social Network Analysis And Mining, [s.l.], p.1511-1522, 2014. Springer New York. http://dx.doi.org/10.1007/978-1-4614-6170-8_88.

[2] SCHWARTZ, Barry. The paradox of choice: Why more is less. New York: Harper Perennial, 2004. 304 p.

[3] WANG, Zan et al. An improved collaborative movie recommendation system using computational intelligence. Journal Of Visual Languages & Computing, [s.l.], v. 25, n. 6, p.667-675, dez. 2014. Elsevier BV. http://dx.doi.org/10.1016/j.jvlc.2014.09.011.

[4] JACKSON, Dan. The Netflix Prize: How a $1 Million Contest Changed Binge-Watching Forever. 2017. Disponível em: <https://www.thrillist.com/entertainment/nation/the-netflix-prize>. Acesso em: 22 ago. 2019.

[5] FERNANDO, Martínez-plumed et al. Making Sense of Item Response Theory in Machine Learning. Frontiers In Artificial Intelligence And Applications, [s.l.], v. 285, n. 2016, p.1140-1148, 2016. IOS Press. http://dx.doi.org/10.3233/978-1-61499-672-9-1140

[6] Chen, Yu, et al. B3-IRT: A New Item Response Model and its Applications. arXiv preprint arXiv:1903.04016, 2019.

[7] HARPER, F. Maxwell et al. The MovieLens Datasets. Acm Transactions On Interactive Intelligent Systems, [s.l.], v. 5, n. 4, p.1-19, 22 dez. 2015. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/2827872.

[8] RICCI, Francesco et al. Recommender Systems Handbook. New York: Springer Us, 2011. 842 p.

[9] CREMONESI, Paolo; KOREN, Yehuda; TURRIN, Roberto. Performance of recommender algorithms on top-n recommendation tasks. Proceedings Of The Fourth Acm Conference On Recommender Systems - Recsys '10, [s.l.], p.39-46, 2010. ACM Press. http://dx.doi.org/10.1145/1864708.1864721.

[10] KOSKELA, Pentti. COMPARING RANKING-BASED COLLABORATIVE FILTERING ALGORITHMS TO A RATING-BASED ALTERNATIVE IN RECOMMENDER SYSTEMS CONTEXT. 2017. 52 f. Dissertação (Mestrado) - Curso de Information Systems Science, JyvÄskylÄn Yliopisto, Seminaarinkatu, 2017.

[11] SPEARMAN, C.. The Proof and Measurement of Association between Two Things. The American Journal Of Psychology, [s.l.], v. 15, n. 1, p.72-101, jan. 1904. JSTOR. http://dx.doi.org/10.2307/1412159.

[12] HUG, Nicolas. Surprise, a Python library for recommender systems. 2017. Disponível em: <http://surpriselib.com>. Acesso em: 05 ago. 2019.

[13] KOREN, Yehuda. Factor in the neighbors. Acm Transactions On Knowledge Discovery From Data, [s.l.], v. 4, n. 1, p.1-24, 1 jan. 2010. Association for Computing Machinery (ACM). http://dx.doi.org/10.1145/1644873.1644874.

[14] KOREN, Yehuda; BELL, Robert; VOLINSKY, Chris. Matrix Factorization Techniques for Recommender Systems. Computer, [s.l.], v. 42, n. 8, p.30-37, ago. 2009. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/mc.2009.263.

[15] LUO, Xin; ZHOU, Mengchu; XIA, Yunni. An Efficient Non-Negative Matrix-Factorization-Based Approach to Collaborative Filtering for Recommender Systems. Ieee Transactions On Industrial Informatics, [s.l.], v. 10, n. 2, p.1273-1284, maio 2014. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tii.2014.2308433.

[16] LEMIRE, Daniel; MACLACHLAN, Anna. Slope One Predictors for Online Rating-Based Collaborative Filtering. Proceedings Of The 2005 Siam International Conference On Data Mining, [s.l.], p.1-5, 21 abr. 2005. Society for Industrial and Applied Mathematics. http://dx.doi.org/10.1137/1.9781611972757.43.

[17] SIMPLER, Data Science Made. How do you build a “People who bought this also bought that”-style recommendation engine. 2015. Disponível em: <https://datasciencemadesimpler.wordpress.com/tag/co-clustering>. Acesso em: 22 nov. 2019.

[18] FUNK, Simon. Netflix Update: Try This at Home. 2006. Disponível em: <https://sifter.org/~simon/journal/20061211.html>. Acesso em: 23 nov. 2019.

[19] EDUCAÇÃO, Ministério da. Teoria de resposta ao item avalia habilidade e minimiza o “chute” de candidatos. 2011. Disponível em: <http://portal.mec.gov.br/institucional/quem-e-quem/389-noticias/ensino-medio-2092297298/17319-teoria-de-resposta-ao-item-avalia-habilidade-e-minimiza-o-chute>. Acesso em: 23 nov. 2019.

[20] GADENZ, Sabrina Dalbosco et al. Elaboração e validação de uma medida para avaliar o conhecimento de médicos de atenção primária do Brasil sobre recomendação nutricional para controle da hipertensão. Cadernos Saúde Coletiva,

[s.l.], p.1-8, 25 nov. 2019. FapUNIFESP (SciELO). http://dx.doi.org/10.1590/1414-462x201900040205.

[21] HARRIS, Deborah. Comparison of 1-, 2-, and 3-Parameter IRT Models. Educational Measurement: Issues and Practice, [s.l.], v. 8, n. 1, p.35-41, mar. 1989. Wiley. http://dx.doi.org/10.1111/j.1745-3992.1989.tb00313.x.

[22] BAKER, Frank B.. The Basics of Item Response Theory. 2. ed. Washington: Eric, 2001.