Universidade Federal de Pernambuco
Graduação em Ciência da Computação
Centro de Informática

FLÁVIO DE HOLANDA CAVALCANTI JÚNIOR

Avaliação de Técnicas de Filtragem Colaborativa para Sistemas de Recomendação

RECIFE
2017

FLÁVIO DE HOLANDA CAVALCANTI JÚNIOR

Avaliação de Técnicas de Filtragem Colaborativa para Sistemas de Recomendação

Trabalho de Conclusão de Curso apresentado para o curso de Ciência da Computação da Universidade Federal de Pernambuco (UFPE) para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Ricardo Bastos C. Prudencio

RECIFE
2017

Resumo

Com a explosão de informações da era digital, principalmente após o advento da Web, usuários encontram-se constantemente expostos a uma imensa variedade de informações, das quais apenas uma pequena fração têm a relevância adequada as suas reais necessidades. Torna-se necessário descobrir meios de filtrar essa grande quantidade de conteúdo, delegando ao usuário apenas as informações (potencialmente) relevantes, ao encontro de seu perfil e/ou necessidade "individual". Desse tipo de problema vêm tratar os “sistemas de recomendações”, encarregado de sugerir, a usuários, itens de seu provável interesse com máxima eficácia. Esses sistemas podem empregar diversas técnicas diferentes, das quais analisaremos três de um nicho específico da área denominada “filtragem colaborativa”. Ao final do documento faremos alguns experimentos e analisaremos seus resultados; as técnicas a serem abordadas serão: user-user, item-item e SVD.

Palavras-chave: Sistema de Recomendação, SVD, Filtragem Colaborativa.

Sumário

1 Introdução

Com a explosão de informações da era digital, principalmente após o advento da Web, usuários encontram-se constantemente expostos a uma imensa variedade de informações, das quais apenas uma pequena fração têm a relevância adequada as suas reais necessidades. Estima-se que a quantidade de dados armazenada em bancos de dados ao redor do mundo é duplicada a cada 20 meses [8], tornando essencial aprimorar-se técnicas de filtragem de grandes quantidades de conteúdo, almejando refinar esses dados e delegar ao usuário apenas as informações (potencialmente) relevantes, ao encontro de seu perfil e/ou necessidade "individual". Outra potencialidade desse tipo de serviço vem do interesse empresarial, que percebem ao disponibilizar recomendações direcionadas ao usuário uma maior probabilidade de aceitação das ofertas, aumentando em consequência sua receita em decorrência do fenômeno de “cauda longa”, que afirma que itens de menor popularidade, quando vendidos ao seu público-alvo, mesmo sendo esse de menor proporção em relação ao qual são destinados os itens considerados populares, conseguem superar a vazão dos ditos itens populares [3]. As empresas vêm se adequando ao conceito de que não basta fornecer um único serviço padrão a todos os usuários, mas múltiplos serviços, que satisfaçam múltiplas necessidades de múltiplos usuários [1], a intenção está em personalizar serviços ofertados em função das necessidades/preferências de cada um de seus clientes/usuários, um princípio bem colocado em uma afirmação feita pelo CEO da Amazon, Jeff Bezos, que certa vez disse: “*If I have 2 million customers on the Web, I should have 2 million stores on the Web*” [1], traduzido como "Se eu tenho 2 milhões de clientes na Web, eu deveria ter 2 milhões de lojas na Web" (tradução nossa).

Um tipo de ferramenta que propõe resolver essa questão da sobrecarga de informação é o chamado Sistema de recomendação [3, 4]. São ferramentas capazes de providenciar recomendações de “itens” a usuários de uma forma personalizada, isto é, de sugerir os itens mais prováveis de satisfazerem interesses individuais do usuário de forma prioritária aos demais, ou seja, itens considerados interessantes são sugeridos ao usuário com maior destaque em detrimento dos "menos" interessantes. Um sistema de recomendação pode sugerir, por exemplo, aquisições de itens como livros, músicas, notícias, etc. para um usuário específico com base em suas preferências e restrições pessoais, características essas que, por sua vez,

poderiam ser obtidas de forma explícita (através de feedback voluntário do usuário) ou implícita (monitorando a atividade do usuário durante a utilização do sistema). Um meio bastante comum de se encontrar aplicações de sistemas de recomendação são os serviços de e-commerce, como no portal online da Amazon.com, auxiliando o usuário no fornecimento de recomendações das (possíveis) melhores aquisições de produtos de seu interesse, em acordo com o seu perfil ou em decorrência do histórico de produtos já adquiridos pelo usuário; pode-se observar também a aplicação de técnicas de recomendação em serviços de outras empresas famosas como: Yahoo, Google, Netflix, eBay, etc. [1, 2, 6, 11]. Estima-se que 60% das visualizações do YouTube sejam decorridas de recomendações do serviço [2].

As técnicas utilizadas por sistemas de recomendação podem ser dissociadas em três vertentes: filtragem colaborativa, filtragem baseada em conteúdo, e filtragem por abordagem híbrida. Na filtragem colaborativa recomendações de itens ao usuário são feitas levando em consideração os interesses de "usuários semelhantes" ao desejado; na filtragem baseada em conteúdo itens são sugeridos com base em "propriedades" ou "conteúdos" semelhantes aos de itens anteriormente avaliados positivamente pelo usuário; nos sistemas híbridos há uma mesclagem das duas classificações anteriores, tentando conciliar suas vantagens e mitigar desvantagens [1, 2, 6].

Em outubro de 2006 foi iniciada uma competição chamada "Netflix Prize" que foi um grande marco, um ponto de inflexão no volume de progressos no campo de filtragem colaborativa. Pela primeira vez a comunidade de pesquisa teve acesso a um dataset de larga escala, contendo 100 milhões de ratings de filmes, que atraiu diversos entusiastas à área. Todos os participantes tinham como objetivo ultrapassar em 10% a precisão (via RMSE) do sistema oficial da Netflix à época, o Cinematch [3, 7, 10, 15, 18]. A competição terminou em 2009, tendo como vencedor o time "Bellkor's Pragmatic Chaos", ganhador do prêmio de 1 milhão de dólares [3, 7, 10, 15, 17].

Analisaremos no decorrer deste documento três técnicas de recomendação na vertente de filtragem colaborativa, comumente aplicadas em sistemas de e-commerce: user-user, item-item e SVD. Escolhemos as duas primeiras por serem técnicas tradicionais de recomendação, normalmente também utilizadas como baselines em comparações a demais outras; a escolha da técnica SVD vem da grande popularidade que obteve após seu relativo sucesso na competição da Netflix Prize, tornando-se desde então uma abordagem bastante comum nos dias atuais após sucessivas melhorias [17, 18]. Ao final do trabalho faremos uma

experimentação comparativa sobre uma base de dados de teste para analisar os resultados dessas técnicas.

Para entender melhor o conceito de Sistemas de Recomendaçao foi dedicado o capítulo 2 deste trabalho a esse tema, explicando ainda algumas de suas ramificações, vantagens e desvantagens.

No capítulo 3 vemos uma introdução ao conceito de Filtragem Colaborativa, a abordagem utilizada por todas as técnicas que serão estudadas neste trabalho.

No capítulo 4 vemos a abordagem Memory-based de Filtragem Colaborativa e duas das suas ramificações, “recomendações baseadas em item” e “recomendações baseadas em usuário”, técnicas que serão utilizadas como baselines nas nossas experimentações do capítulo 6.

No capítulo 5 vemos a abordagem Model-based de Filtragem Colaborativa e um pouco mais sobre a técnica SVD, também em uma versão incremental, que será explorada nas experimentações do capítulo 6.

No capítulo 6, explicamos os experimentos realizados e resultados obtidos ao decorrer deste trabalho, apresentando os parâmetros experimentais, ferramentas, dataset utilizados e técnicas abordadas.

No capítulo 7 fazemos uma breve conclusão do conteúdo deste documento e citaremos possíveis trabalhos futuros que poderiam levantar melhores resultados de nossos experimentos.

2 Sistemas de Recomendação

Sistemas de recomendação são ferramentas que sugerem a usuários “produtos ou serviços” que lhe possam servir algum interesse, não sendo estritamente necessária a esse fim a interação direta com o usuário na especificação dos parâmetros que influenciem essa sugestão ou mesmo a solicitação da sugestão em si. Uma representação comum de recomendação é uma lista de itens estruturada, designada a evidenciar as melhores opções dos itens disponibilizados e, consequentemente, ofuscar os de menor relevância, expondo o usuário majoritariamente a itens de seu provável interesse. O conceito de sistemas de recomendação nasceu de dois princípios: reuso de informação e persistência de preferências. A ideia ligando esses princípios lembra o conceito de “navegação social”, o de que atitudes de indivíduos em um mesmo espaço de informações são influenciadas por atitudes de outros no mesmo espaço. A ideia, portanto, é a de que usuários costumam melhor identificar-se com aqueles de comportamentos/preferências similares e, sendo essas preferências em geral constantes (não imutáveis), como costumam ser, ao registra-las formamos um perfil dos usuários nos habilitando a estipular suas similaridades e prever possíveis comportamentos futuros tomando como base seus pares [11, 13].

Um exemplo comum e simples de aplicação de sistemas de recomendação é visto em portais de e-commerce e streaming em quais listas de "itens recomendados" são exibidas ao usuário cujo cálculo efetuado na geração do ranking (pontuação que define o destaque a ser dado ao item na representação da recomendação) é uma simples agregação estatística (itens “mais vistos”, “mais compartilhados”, “mais procurados”, “quantidade de marcações como favoritos”, etc.). Esses tipos de sistemas compõem o gênero mais simples de sistema de recomendação existentes - **sistemas de recomendação não-personalizados não-contextualizados** - que, como nos exemplos citados, levam em conta apenas estimativas globais de algum atributo, normalmente no intuito de simular a boa aceitação de um item para uma parcela majoritária dos clientes (visualizações, compartilhamentos, etc.) assumindo que “todos” os usuários são similares entre si, não levando em consideração portanto interesses pessoais - os itens recomendados são os mesmos para qualquer usuário - ou qualquer outro contexto em que este se enquadre e que os diferencie. Esse tipo de sistema de recomendação é o também comumente encontrado em lojas físicas, em quais produtos com maior “saída” são

rotineiramente dispostos em destaque, expondo todos os clientes da loja a uma percepção única, compartilhada, dos produtos ali oferecidos, não oferecendo, portanto, serviços ou recomendações personalizados para qualquer usuário nesse sentido [1, 11].

Existem também **sistemas de recomendação não-personalizados contextualizados**, isto é, embora suas recomendações continuem não expressando interesses pessoais do usuário, elas buscam expressar o "estado" atual desse usuário ao levar em conta, por exemplo, características como momento, localização, e ambiente atuais, i.e., informações externas ao usuário, às quais está submisso, e que poderiam "influenciar" suas escolhas (data, horário, clima, temperatura, local, etc.). Um usuário, por exemplo, pode estar mais propenso a procurar itens natalinos no mês de dezembro, locais abertos e adereços compatíveis durante climas quentes, encontrar petiscos ou filmes estreantes do mês ao entrar em um cinema, etc. Pode-se levar em conta ainda nas recomendações as interações que o usuário venha a ter com o sistema, como por exemplo cliques e avaliações ou adições de itens a sua "cesta de compras" na representação do contexto do usuário [11].

Sistemas não-personalizados contextualizados são comuns em sites de e-commerce em que há costume de recomendar itens relacionados aos existentes na "cesta de compras" do usuário. O cálculo da correlação entre conjuntos de itens nesses sistemas é normalmente pensado em função da frequência com que, dado um conjunto X de itens previamente consumido, um conjunto Y de itens ser consumido futuramente; pode-se levar em conta apenas dados entre transações únicas ou entre agregações, ou seja, agrupamentos de transações por dias, semanas, etc.. Algumas fórmulas que costumam tratar esse problema são mencionadas abaixo [11]. Um meio concreto de se calcular a correlação seria através da própria função conceitual, ou seja, o cálculo da frequência com que todos os itens em $X \cap Y$ – todos os elementos presentes em X e em Y – são consumidos conjuntamente, em razão da frequência com que "pelo menos" todos os itens em X são consumidos, i.e.:

$$\frac{P(X \cap Y)}{P(X)} = P(Y|X) \qquad (1)$$

O problema com essa abordagem é que não é levada em conta a popularidade do conjunto Y. Isso possivelmente levaria a uma falsa correlação entre X e Y, pois sendo Y popular esse estaria presente em grande parte das transações, aumentando a chance de se encontrá-lo em transações portando X, ou seja, aumentando a frequência de transações

contendo $X \cap Y$. Um modo de adaptar essa fórmula para levar em conta a popularidade de Y seria dividindo seu resultado por $P(Y|\bar{X})$; esse denominador garante que sendo Y popular, ou seja, sendo $P(Y|\bar{X})$ alto, o resultado da fórmula seguirá reduzido, independente do numerador. Note que o contradomínio dessa nova fórmula passa a ser $[0,\infty]$, não mais $[0,1]$. Teríamos então a seguinte fórmula como nova métrica:

$$\frac{\frac{P(X \cap Y)}{P(X)}}{\frac{P(\bar{X} \cap Y)}{P(\bar{X})}} = \frac{P(Y|X)}{P(Y|\bar{X})}$$

Note que ambas as métricas citadas acima são "direcionadas", isto é, os resultados que buscam a frequência de um "conjunto de itens Y para um dado conjunto X previsto" é (provavelmente) diferente do obtido quando os parâmetros estão invertidos, isto é, de um "conjunto de itens X para um dado conjunto Y previsto". Há métricas de correlação, no entanto, "não-direcionadas", ou seja, a ordem dos parâmetros não altera seu resultado. Um exemplo dessas métricas é o *lift*, cujo cálculo é dado pela razão entre a fórmula (1), i.e., $P(Y|X)$ e a probabilidade de se obter todos os itens em Y. Veja a equação abaixo, que exibe três notações diferentes para a métrica *lift*, e note que a troca dos parâmetros X e Y não altera o resultado final, justificando-lhe a atribuição do termo "não-direcionada". Outra possível interpretação, também visível pela equação abaixo, se dá pela razão entre a probabilidade real com que os elementos em X e Y são encontrados em uma mesma transação, e a probabilidade com que os mesmos seriam encontrados "se X e Y fossem independentes entre si", isto é, se a aquisição de X não influenciasse a aquisição de Y, e vice-versa; logo, não há correlação entre os conjuntos quando esta métrica tem resultado 1 [2, 11]. Note que, assim como na métrica anterior, esta fórmula tem contradomínio no intervalo $[0,\infty]$.

$$\frac{P(X \cap Y)}{P(X)P(Y)} = \frac{P(X|Y)}{P(X)} = \frac{P(Y|X)}{P(Y)}$$

É comum estabelecer um *threshold*, um limiar, abaixo do qual o resultado de uma métrica de correlação será considerado "insuficiente" para estabelecer uma conexão significativa entre os grupos de itens considerados. Logo, mesmo valores acima de 1, via de regra indicadores de correlação positiva pela métrica *lift*, poderiam ser ignorados por não constituírem uma representação significativa de correlação havendo um *threshold* nesse

sentido [2]. Há ainda outros meios de se encontrar correlações entre conjuntos de itens de forma não-personalizada que não serão abordados neste trabalho por não comporem o seu objetivo principal.

Por fim, há também **sistemas de recomendação personalizados**, cujas recomendações são focadas no perfil do usuário, indo além das estatísticas globais ou contexto de usuário, trabalhando com o histórico de interações do usuário com o sistema e seu registro de preferências. Há três subtipos gerais de sistemas de recomendação personalizados em razão da abordagem na elaboração das recomendações:

1. Filtragem Colaborativa [12, 18] – emprega o conceito de que usuários pertencentes a um grupo de pessoas "semelhantes" tendem a melhor aceitar opiniões populares do seu círculo do que de pessoas externas – opiniões dessas podem ser consideradas algo aleatório ou estereotipado por uma maioria com quem o usuário não se identifica. Para expressar esse conceito a filtragem colaborativa calcula recomendações extraindo grupos de usuários similares a partir dos seus históricos de comportamentos, descobre os itens mais populares em cada grupo e recomenda a usuários de cada grupo apenas os itens mais populares em cada que ainda não tenha sido consumido pelo usuário.
2. Filtragem Baseada em Conteúdo [12, 18] – emprega o conceito de que a preferência do usuário por um item está associada a determinadas características pelas quais aquele item pode ser definido e que, portanto, seria possível migrar a associação entre usuários e itens para uma associação entre usuários e "características" relevantes comuns aos itens disponíveis. Seria possível então, a partir dessa generalização, prever quais outros itens o usuário estaria propenso a consumir, correlacionando-se as características do item às por quais o usuário tem preferência. Note que, ao contrário da filtragem colaborativa, não é levado em conta as preferências de outros usuários do sistema.
3. Filtragem Híbrida [12, 18] – emprega tanto o conceito de filtragem colaborativa como o de filtragem baseada em conteúdo. A intenção nesse tipo de sistema é unir os benefícios e mitigar os problemas encontrados em cada abordagem, quando analisados isoladamente, gerando assim melhores recomendações finais ao usuário.

Um analista, Jack Aaronson, do Aaronson Group, estimou que investimentos em sistemas de recomendações retornam em 10 a 30 por cento do lucro habitual de uma empresa [14]. Entre outros benefícios decorrentes da adoção de um sistema de recomendação estão:

1. Aumento do número de itens adquiridos [1, 11] – o principal objetivo de todo sistema de recomendação é o aumento do número de aceitações das suas recomendações. Um maior número dessas aceitações indica uma maior eficácia do sistema em gerar consumo por parte de seus clientes. Na vida real é comum que uma pessoa esteja mais propensa a adquirir um item se este lhe foi recomendado por um amigo ou alguém a quem respeita; a mesma coisa acontece no contexto de sistemas recomendações, em que o sistema toma o papel do amigo, buscando criar laços de confiança com o usuário, tentando prover recomendações relevantes e torná-lo mais propenso a aceitar recomendações futuras. Itens recomendados dessa forma, de acordo com o perfil do usuário, se tornam também mais visíveis, expõem o usuário a alternativas interessantes que de outra forma não poderiam ser consideradas por não serem referências, não conhecê-los ou por prejulgamentos, por exemplo;
2. Facilitar a descoberta de itens similares (*cross-selling*) [1] – ao recomendar "itens similares" a um item-alvo previamente adquirido pelo usuário ou que esse demonstre a intenção de adquirir o serviço tende a aumentar a quantidade de itens consumidos. Itens podem ser determinados similares por vários aspectos, como finalidade, gênero, qualquer outra propriedade em comum com o item-alvo ou mesmo por tratar-se de um item costumeiramente adquirido conjuntamente ao item-alvo;
3. Facilitar a descoberta e aumentar o consumo de itens "impopulares" – itens impopulares são recomendados apenas a pessoas de perfil compatível, portanto aumentando a quantidade de aquisições desses. Da mesma forma, evita a sugestão de itens impopulares a pessoas erradas, sem o perfil adequado. Assim, ambos os perfis (adequados ou não a estes itens impopulares) saem beneficiados;
4. Ofertas [8] - se o sistema de recomendação detectar que um item em que o usuário demonstra interesse apresenta um histórico de aquisições conjugadas a um determinado outro item, e que este ainda não foi adquirido pelo usuário, o sistema pode oferecer um desconto/cupom na aquisição conjunta desses dois itens. Ofertas conjuntas se baseiam na suposição de que clientes, ao sentir-se beneficiados em uma

transação, mesmo que esta não lhes seja necessária, apresentam maiores chances de a adquirirem;

5. Aumentar satisfação e fidelidade do usuário com o sistema [1] – um usuário ao se deparar com um sistema personalizado que atende as suas necessidade tende a utilizar o sistema com maior frequência, em detrimento de outros sistemas concorrentes, resultando numa maior quantidade de itens adquiridos via esse sistema. Outro fator que eleva a fidelidade do usuário com o sistema está no próprio período de uso do sistema de recomendação, pois esses sistemas costumam tornar-se mais precisos à medida em que são mais utilizados, pois o histórico de aquisições/avaliações de itens, interesses, etc. torna-se mais extenso. Isso faz com que o usuário fique mais avesso a deixar o sistema atual por um outro, já que teria de passar por um novo processo de treinamento no outro sistema para obter recomendações aceitáveis, possivelmente levando um longo tempo até conseguir um histórico de avaliações, comportamentos, aquisições, etc. equiparável ao do sistema atual;
6. Aproximação maior com o usuário, e melhor compreensão de suas vontades – ao coletar o feedback, implícito e/ou explícito, dos usuários, o provedor do sistema de recomendação pode redirecionar esse conhecimento para outras frentes da organização, aprimorando por exemplo o gerenciamento de estoque e produção, marketing, etc.

Neste trabalho abordaremos em mais detalhes apenas técnicas de filtragem colaborativa; trataremos desse assunto em mais detalhes no próximo tópico.

3 Filtragem Colaborativa

Uma abordagem de sistemas de recomendação bastante utilizada em aplicações de *e-commerce* é a de "filtragem colaborativa". O princípio dessa abordagem está em fundamentar suas recomendações no pressuposto de que usuários de preferências ou hábitos historicamente similares continuarão seguindo padrões de comportamento similares – remetendo-nos ao conceito de "navegação social" mencionado anteriormente. A ideia geral por trás da filtragem colaborativa está em extrair grupos de usuários semelhantes, mediante históricos de seus

comportamentos, e gerar recomendações para um determinado usuário embasadas nos comportamentos de outros usuários em seu grupo.

O conjunto de entrada de dados para uma técnica de filtragem colaborativa é geralmente representado em triplas de valores. Tem-se em cada tripla as informações da avaliação (valor numérico ou ordinal), doravante chamado de rating, identificada pelo usuário que a fez a um item específico do domínio em questão (filmes, músicas, etc.). Logo, o histórico de comportamentos necessário ao cálculo da similaridade entre usuários, em sua forma mais simples, tem uma tripla de estrutura <*id_usuário, id_item, rating_usuário_item*>, não necessariamente nesta ordem. Note que, a partir desses dados, é possível gerar uma matriz, denominada "matriz usuário-item" (ou user-item), tal que os valores de cada célula da matriz é o valor do rating de um usuário a um determinado item, correspondentes a linha e coluna da célula, ou vice-versa.

Alguns dos problemas mais comumente encontrados em técnicas de filtragem colaborativa são [10]:

- Esparsidade dos dados (*Data Sparsity*): Em sua maioria, um banco de dados contendo registros de avaliações feitas por usuários a itens do domínio pode ser visualizado como uma imensa matriz esparsa de avaliações, isto é, apenas uma pequena fração dos itens disponíveis se encontram avaliados por um usuário, assim como também apenas uma fração de todos os usuários disponíveis avaliam um determinado item;
    - *Cold Start*: Ao adicionar um novo usuário ou item ao domínio do sistema não há de início meios de lhe serem geradas recomendações, visto que não há avaliações pregressas em quais fundamentar-se, a fim de encontrar usuários, ou itens, semelhantes e daí extrair-lhes recomendações. Esse problema costuma perdurar mesmo enquanto o usuário/item possui apenas poucas avaliações realizadas, pois estas podem mostrar-se inviáveis para extrair preferências suficientemente precisas e estáveis a fim lhe direcionar a um grupo adequado, do qual se possa estipular recomendações relevantes.
- Escalabilidade: Sistemas que empregam técnicas de recomendação costumam ter seu banco de dados atualizados continuamente, seja via inserção de novas avaliações por usuários ou atualizações de avaliações pregressas. Essa mudança constante nos perfis dos usuários exige, em consequência, uma contínua atualização nas recomendações que lhes serão feitas, o que pode sobrecarregar a aplicação visto que cada

recomendação é produto de uma análise comparativa entre todos os itens candidatos à recomendação via suas avaliações individuais já fornecidas; assumindo-se que um banco de dados pode conter dezenas de milhões de usuários e milhões de itens em seu catálogo, e que este número é em geral crescente, esta não pode ser considerada uma tarefa simples;

- Sinônimos: há situações em que um mesmo item, ou itens muito idênticos, têm representações distintas em um mesmo sistema, isto é, são referenciados por identificadores, nomes, diferentes. Esse fenômeno causa uma perda de performance no algoritmo pois as similaridades desses sinônimos com os demais itens do banco de dados podem ser diferentes umas das outras, ou seja, alguns itens podem ou não ser recomendados ao usuário, variando em acordo com qual dos sinônimos o cálculo do ranking da recomendação foi baseado. Abordagens baseadas em conteúdo não apresentam este tipo de problema, visto que suas recomendações são calculadas com base nas características comuns inerentes aos produtos;
- *Shilling Attacks*: Prática realizada por usuários mal-intencionados que buscam, por interesses próprios, promover determinados itens e/ou dissuadir usuários de consumir outros itens, o fazendo através de suas próprias avaliações, avaliando positivamente itens de seu interesse e negativamente itens contrários aos seu propósitos (dos possíveis concorrentes);
- *Gray Sheep*: Refere-se a usuários cujas avaliações não têm uma representação significativa com quaisquer um dos grupos de usuários disponíveis e, portanto, não podem receber recomendações.

As técnicas de filtragem colaborativa, por sua vez, podem ser estratificadas em duas outras classes distintas [7, 10, 11, 15, 17, 18]:

1. Baseado em memória (*memory-based* ou *neighborhoood-based*): recomendações computadas com base na matriz user-item, completa ou amostral, de ratings; a representação da matriz é mantida em memória. Nesta abordagem, cada usuário é parte de um grupo de usuários com interesses afins, denominados “vizinhos” – ao identificar essa similaridade entre usuários torna-se possível recomendar-lhes novos itens. Este tipo de técnica é notavelmente empregado em sistemas comerciais por serem eficazes e fáceis de implementar.

2. Baseado em modelo (*model-based* ou *latent factor model*): utilizam ratings dos usuários para construir um "modelo" capaz de fazer predições. Esses modelos são capazes de representar características latentes de usuários e itens, normalmente construídos via técnica de aprendizagem de máquina.

4 Memory-based

Memory-based é uma abordagem de filtragem colaborativa cujas recomendações são computadas via matriz user-item, completa ou amostral, de ratings, necessitando manter essa estrutura em memória, e pode ser efetuada por duas vertentes: "baseada em usuário" ou "baseada em item" [18], explicadas a seguir. Os algoritmos que dão origem a essas recomendações utilizam técnicas baseadas em vizinhança, isto é, elaboram predições de ratings de um item a um determinado usuário através da similaridade entre esse usuários e os demais disponíveis no dataset – recomendação baseada em usuários - ou na similaridade do item em questão com os demais disponíveis no dataset - recomendação baseada em itens. Quanto mais similar (próximo) estiver um usuário ao usuário alvo da predição sendo calculada maior será sua influência no resultado final – o mesmo conceito aplica-se para a similaridade entre itens. Em alguns casos, no entanto, o antagonismo (distância excessiva) entre usuários/itens também pode ser levado em conta no cálculo das predições [11, 15, 16].

Para explanar conceitos futuros definiremos algumas notações abaixo. Por vezes usaremos a notação "usuário ativo" e "item ativo" como referência ao usuário e item, respectivamente, aos quais se deseja calcular uma predição:

- I: Conjunto de todos os itens presentes no dataset;
- U: conjunto de todos os usuários presentes no dataset;
- R: Matriz user-item de todos os ratings definidos por usuários em U a itens em I.

Para itens i e j pertencentes a I, e usuários u e v pertencentes a U, usaremos também as seguintes notações:

- r_i: Vetor pertencente ao espaço vetorial $\mathbb{R}^{|U|}$, cujas coordenadas são todos os (possíveis) ratings fornecidos ao item i. Observe que este vetor inclui coordenadas com ratings não definidos, isto é, quando o item i não é avaliado por usuários em U;

- r_u : Vetor pertencente ao espaço vetorial $\mathbb{R}^{|I|}$, cujas coordenadas são todos os (possíveis) ratings provenientes do usuário u. Observe que este vetor inclui coordenadas com ratings não definidos, isto é, quando itens em I não são avaliados pelo usuário u;
- I_u : Conjunto de itens em I que são avaliados pelo usuário u;
- U_i : Conjunto de usuários em U que avaliam o item i;
- I_{uv} : Conjunto de itens em I que são avaliados conjuntamente pelos usuários u e v;
- U_{ij} : Conjunto de usuários em U que avaliam conjuntamente os itens i e j;
- r_{ui} : Rating avaliado pelo usuário u ao item i;
- $\hat{r}_{ui}$: Predição de rating avaliado pelo usuário u ao item i.

4.1 Recomendação baseada em Usuários

Na recomendação baseada em usuários a predição feita para um determinado item i a um usuário u é estipulada com base na similaridade dos ratings desse usuário com os de todos os demais em U_i, ou seja, de todos os usuários que já tenham avaliado o item i e que, portanto, possam lhe embasar uma opinião – por ora abstrairemos a forma como essa similaridade é computada, voltaremos a este conceito nos tópicos 4.3 e 4.4. Dizemos então que, para todo usuário v em U_i, são considerados vizinhos de u aqueles cujos ratings em I_{uv} estão melhor correlacionados aos seus equivalentes providos por u. Normalmente é definida uma constante k para delimitar o número máximo de vizinhos do usuário que serão levados em conta no cálculo da predição [3, 10, 18] – usaremos a notação $N_i(u)$ para indicar esse subconjunto de vizinhos de tamanho k.

Uma vez calculadas as similaridades entre o usuário u e cada usuário v em U_i, denotadas por w_{uv}, pode-se calcular o valor final da predição $\hat{r}_{ui}$ atribuindo-se pesos a cada um dos ratings fornecidos pelos vizinhos de u ao item i, $N_i(u)$, tal que seus pesos são essas próprias similaridades computadas; então, para que a soma desses ratings, em valor absoluto, não extrapole o intervalo de ratings permitidos, dividimos esse resultado pela soma dos valores absolutos das similaridades [10, 18], ou seja:

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

Um problema com esta abordagem é que normalmente usuários costumar ter escalas de ratings subjetivas, isto é, para uma escala de ratings oficial com valores variando de 1 a 5, em ordem crescente de afinidade, um usuário v pode preferir dar ratings 5 apenas a itens excepcionais e avaliar com rating 3 os demais itens considerados apenas bons; já um usuário x pode ser menos criterioso e avaliar com ratings altos, 4 e 5, por exemplo, a maioria dos itens de sua preferência. Para evitar esses biases subjetivos dos usuários costuma-se normalizar os ratings, amortizando essas limitações na escala de rating que terminam por gerar ruído ao conjunto de dados analisado. Esse tipo de pré-processamento pode ser necessário se levado em conta que o propósito da recomendação é identificar o "grau de afinidade" expressado por um usuário a um item e que, na forma apresentada, eles mostram-se enviesados devido aos comportamentos idiossincráticos de cada usuário. Para expressar esse novo conceito, alterando a fórmula anterior, tomaremos por h a função que normalizará o rating, e por h^{-1} a sua função inversa. A nova fórmula para se calcular a predição $\hat{r}_{ui}$ é então dada por [18]:

$$\hat{r}_{ui} = h^{-1} \frac{\sum_{v \in N_i(u)} w_{uv} h(r_{vi})}{\sum_{v \in N_i(u)} |w_{uv}|}$$

Neste trabalho tomaremos a função de normalização como a de centralização pela média, ou seja, $h(r_{ui}) = r_{ui} - \bar{r}_u$, logo a equação anterior toma a seguinte forma [10, 18]:

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

4.2 Recomendação Baseada em Itens

A recomendação baseada em itens segue a mesma lógica da recomendação baseada em usuários [3, 18], alternando, no entanto, o foco para a similaridade entre "itens" ao invés de entre "usuários". Na recomendação baseada em itens a predição feita para um determinado item i a um usuário u é estipulada com base na similaridade dos ratings do item i com os de todos os demais em I_u, ou seja, de todos os itens já avaliados por u que, portanto, supõem uma orientação quanto às preferências individuais do usuário. Dizemos então que, para todo item j em I_u, são considerados vizinhos de i aqueles cujos ratings fornecidos por U_{ij} estão melhor correlacionados aos seus equivalentes providos por i. Assim como na abordagem orientada a usuário, normalmente é definida uma constante k para delimitar o número máximo de vizinhos do item que serão levados em conta no cálculo da predição – usaremos a notação $N_u(i)$ para indicar esse subconjunto de vizinhos de tamanho k. Portanto, a fórmula equivalente para a recomendação baseada em itens é:

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in N_u(i)} w_{ij}\left(r_{uj} - \bar{r}_j\right)}{\sum_{j \in N_u(i)} |w_{ij}|}$$

Existem vários modos de se computar a similaridade utilizada nos cálculos da predição; veremos a seguir dois desses métodos tradicionalmente utilizados em sistemas de recomendação [18]: "Cosseno similaridade" e "Correlação de Pearson".

4.3 Cosseno Similaridade

Na "Cosseno similaridade" expressamos a similaridade como a proximidade entre os vetores dos ratings, quer sejam esses vetores relacionados a um item (recomendação baseada em itens) ou a um usuário (recomendação baseada em usuários). Para efeito demonstrativo durante a explanação usamos a abordagem baseada em usuários, embora sua correspondente baseada em itens seja feita de modo equivalente. Para vetores de ratings r_u e r_v relacionados aos usuários u e v, respectivamente, a métrica "cosseno similaridade" determina o cosseno do ângulo formado entre esses vetores [3, 6, 10, 16, 18]:

$$\cos(r_u, r_v) = \frac{r_u^T r_v}{\|r_u\|\|r_v\|} = \frac{\sum_{i \in I_{uv}} r_{ui} r_{vi}}{\sqrt{\left(\sum_{i \in I_{uv}} r_{ui}^2\right)\left(\sum_{j \in I_{uv}} r_{vj}^2\right)}}$$

Note que, por representar o cosseno entre vetores, o momento em que os vetores de ratings estão mais próximos, demonstrando máxima similaridade entre si, está quando formam entre si um ângulo de 0º, isto é, quando seu cosseno tem valor 1. Na mesma lógica, apresentam maior rejeição entre si quando formam um ângulo de 180º, resultando em um cosseno de -1. Vemos então que a cosseno similaridade gera valores no intervalo $[-1, 1]$, tal que -1 representa a máxima incompatibilidade entre os usuários – o usuário u tem aversão às opiniões de v, e vice-versa –, o valor 0 representa completa indiferença entre os usuários, ou seja, opiniões de u não influenciam comportamentos de v, e vice-versa, e o valor 1 representa a máxima similaridade entre os usuários (idênticos).

Note também que se os ratings forem valores não-negativos o intervalo de resultados da fórmula passa a ser $[0,1]$, não sendo mais possível exprimir a aversão de opiniões entre usuários.

4.4 Correlação de Pearson

Como já visto, basear-ser nos valores reais dos ratings pode não ser uma boa ideia já que usuários/itens podem podem apresentar biases em seus ratings. Pode-se dizer então que os vetores de ratings têm dois componentes, o bias inerente ao usuário/item, que aqui tomaremos como a média de seus ratings, e a sua variação da média, i.e. $r_u = r'_u + \bar{r}_u$, onde o termo r'_u é o próprio vetor r_u centralizado em relação a sua média $\bar{r}_u$.

$$\cos(r_u - \bar{r}_u, r_v - \bar{r}_v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\left(\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2\right)\left(\sum_{j \in I_{uv}} (r_{vj} - \bar{r}_v)^2\right)}}$$

Note que ratings não definidos para um determinado item, ou seja, itens não avaliados quer seja pelo usuário u ou v, não são levados em conta na função, por se limitar apenas o domínio dos itens avaliados “conjuntamente” por ambos usuários. Essa métrica é denominada

"Correlação de Pearson", também expressada como a razão entre a covariância dos vetores r_u e r_v e o produto de seus desvios padrões [10, 16, 18]:

$$cor(r_u, r_v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\left(\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2\right)\left(\sum_{j \in I_{uv}} (r_{vj} - \bar{r}_v)^2\right)}} = \frac{cov(r_u, r_v)}{\sigma_{r_u} \times \sigma_{r_v}}$$

A principal distinção desta métrica em relação ao cosseno é por promover a eliminação de alguns biases através da remoção das diferenças na média e variância dos vetores de ratings, expressando a extensão com que esses vetores estão linearmente correlacionados [10, 11, 12]. Note ainda que esta função resulta em valores no intervalo $[-1, 1]$, tal que o valor 1 indica uma correlação perfeita positiva entre os vetores, -1 uma correlação negativa perfeita, e o valor 0 exprime a inexistência de correlação linear entre eles.

Por muito tempo a abordagem em vizinhança foi o padrão adotado para sistemas de recomendação baseados em filtragem colaborativa, que só passou a ser contrariado quando surgiram técnicas mais eficazes de redução de dimensionalidade model-based [18], como abordado no capítulo a seguir.

5 Model-Based

Abordagens model-based utilizam os ratings dos usuários para construir (treinar) um "modelo" capaz de fazer predições. Esses modelos mapeiam usuários e itens para um espaço compartilhado único através de redução de dimensionalidade, onde ambos estão representados em um mesmo conjunto de "características latentes", extraídos (aprendidos) dos ratings originais, portanto diretamente comparáveis [7, 10, 11, 15, 17, 18].

Entre possíveis características latentes, relacionando usuários a filmes, por exemplo, poderia-se extrair características como gênero, faixa etária, profundidade no desenvolvimento dos personagens, ou mesmo características cujas denominações (ainda) não estão definidas [7].

Abordagens model-based costumam ter precisão melhor do que as memory-based por tratar vários dos aspectos latentes no relacionamento usuário-item, no entanto ao criar essa camada intermediária abstrata de características latentes ela dificulta a explicação das

recomendações ao usuário, algo que a abordagem item-item (memory-based) tem facilidade em demonstrar. Logo, a escolha da abordagem de recomendação a ser usada depende muito da aplicação prática a qual está destinada [7].

5.1 Redução de Dimensionalidade

É comum em sistemas de recomendação trabalhar com datasets com uma grande quantidade de features, i.e., datasets de alta dimensionalidade. Uma consequência comum que acompanha conjuntos de dados nessa composição é a maior esparsidade desses dados na medida em que a quantidade de features aumenta. A isso se dá o fato de que é comum que registros do dataset não necessariamente assumam valores para todas as features disponíveis nele, logo se essa quantidade de features é muito grande uma maior esparsidade torna-se viável.

As noções de densidade e distância entre pontos, críticos para técnicas de clustering e detecção de outliers, tornam-se menos significativas em espaços de alta dimensão [18]. Isso é conhecido como a "Maldição da Dimensionalidade" [13, 18]. Técnicas de redução de dimensionalidade ajudam a superar esse problema transformando o espaço original de alta dimensionalidade em um de menor dimensionalidade; deve-se no entanto atentar ao fato de que na medida em que as dimensões são removidas do conjunto de dados original a precisão dessa aproximação ao conjunto original se torna reduzida [3, 10].

Existem várias técnicas que abordam esse tipo de técnica, entre elas está a "Singular Value Decomposition (SVD)", que é objeto de nosso estudo no tópico seguinte.

5.2 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) é uma técnica de decomposição de matrizes que tem como consequência a redução da dimensionalidade de um dataset. Em suma ela tenta agregar uma grande quantidade de features em um conjunto reduzido de "conceitos" (atribuindo-lhes também algum parâmetro de intensidade), "acepções" mais abrangentes das features de quais foram derivados, conseguindo dessa forma generalizar o conhecimento

obtido e torná-los aplicáveis a casos mais variados. Remove-se dessa forma também muito dos ruídos (outliers) presentes no dataset [3, 18].

O SVD se baseia no teorema de que é possível decompor qualquer matriz A em $A = U\Sigma V^T$. Dada uma matriz $A_{m\times n}$ (m itens, n features), pode-se obter uma matriz $U_{m\times r}$ (m itens, r conceitos), uma matriz diagonal $\Sigma_{r\times r}$ representando a intensidade de cada conceito e uma matriz $V_{n\times r}$ (n features, r conceitos). A matriz Σ contém os valores singulares, que são positivos e ordenados em ordem decrescente [3, 5, 9, 18]. A matriz U é interpretada como a matriz de similaridade item-conceito, enquanto a matriz V a matriz similaridade de termo-conceito [18].

Para chegar a essa decomposição define-se a matriz U como aquela em qual seus vetores colunas são os autovetores de AA^T; a matriz V como aquela em qual seus vetores colunas são os autovetores de $A^T A$; e a matriz diagonal Σ como aquela em qual seus valores são as raízes quadradas positivas dos autovalores de ambos: AA^T e $A^T A$ [5, 18].

É possível pela técnica SVD calcular matrizes aproximadas de A tendo apenas um parâmetro k como variante dessas possíveis matrizes. Isso é facilitado pelo fato dos autovalores em Σ estarem ordenados de maneira decrescente, assim basta truncar tal matriz no k-ésimo autovalor para obter sua aproximação, representada por $A_k = U_k \Sigma_k V_k^T$. A^k é definida como a matriz de rank k "mais próxima" de A [5, 9, 18], levando-se em conta a minimização da soma dos quadrados das diferenças entre elementos de A e A_k - daí a notoriedade da técnica de SVD na ciência para redução de dimensionalidade.

Contextualizando a técnica SVD em um sistema de recomendação pode-se interpretar a matriz A como uma matriz de ratings fornecidos por usuários (linhas da matriz) a itens específicos (colunas da matriz). A técnica SVD, portanto, mapeia ambos, usuários e itens, para um espaço de fatores latentes comum de dimensionalidade k, tornando-os diretamente comparáveis [7]. Considerando itens como músicas, por exemplo, tais fatores mensurariam conceitos como gênero musical, cantor, remix, ritmo, etc.

Os fatores gerados pelo SVD podem ser representados por vetores $q_i \in \mathbb{R}^k$ e $p_u \in \mathbb{R}^k$, para todo item i e usuário u, isto é, os elementos em q_i medem a extensão com que o item i é representado naqueles fatores, e p_u a extensão do interesse do usuário u nos itens de fatores correspondentes. Em ambos os casos os fatores podem ser positivos ou negativos. O produto interno desses vetores, então, resultará no rating do usuário u ao item i: $r_{ui} = p_u q_i^T$. À matriz

reduzida $A_k = U_k \Sigma_k V_k^T = PQ^T$, então, pode-se dizer que $P = U_K \sqrt{\Sigma_K}$ corresponde à matriz de fatores de usuários, e a matriz $Q = V_K \sqrt{\Sigma_K}$ à de fatores de itens [5, 9, 17, 18].

A técnica SVD é uma das técnicas de decomposição de matrizes mais amplamente utilizadas na ciência, bastante reconhecida por sua precisão e escalabilidade no nicho de sistemas de recomendação, em especial por conseguir chegar à matriz de rank k "mais próxima" de A após aplicada a redução de dimensionalidade. Entre as vantagens dessa abordagem em relação às baseadas em abordagem em vizinhança estão [5, 7, 9, 18]:

- O modelo reduzido do SVD ocupa menos espaço que a matriz de similaridades tradicional;
- É melhor em trabalhar com datasets esparsos, já que ao reduzir a dimensionalidade aumenta-se a densidade e diminui-se o ruído dos dados;
- Pode estabelecer sinonímia e relações transitivas entre itens, já que a semelhança do itens não é contada com base em usuários avaliadores "específicos", mas em "conceitos" extraídos deles, logo itens sem muitos avaliadores em comum podem ser considerados similares em uma abordagem SVD, o que não seria de outra forma.

Entre as desvantagens em comparação as abordagens em vizinhança estão [7]:

- Dificuldade em explicar as razões para o resultado de uma predição, devido a camada intermediária de fatores latentes geradas na técnica SVD;
- Novos ratings requerem reaprendizado dos fatores de usuário e itens, portanto um novo processo de decomposição da matriz, que é extremamente custoso.

Embora a técnica SVD seja extremamente útil, seu processo de decomposição é computacionalmente extremamente custoso - $O(m)^3$ para uma matriz $A_{m\,x\,n}$ [5] - para aplicações em sistemas de recomendação que atualizam suas recomendações com base em dados muito "recentes", e com grande frequência, prejudicando o quesito escalabilidade [5, 18]. A necessidade de um algoritmo menos custoso e de precisão similar, que conseguisse trabalhar de forma incremental, foi uma das grandes preocupações principalmente durante a competição "Netflix Prize" [10]. Analisaremos a seguir uma estratégia incremental do SVD.

5.3 Singular Value Decomposition Incremental

Como citado anteriormente, uma abordagem incremental do SVD se faz necessária para satisfazer melhores resultados de desempenho, principalmente para sistemas de recomendação em tempo real cuja base de dados adjacente é constantemente atualizada.

Para entendermos o modo incremental vamos considerar um conjunto $K=\{(u,i)|r_{ui}\,é\,conhecido\}$ e levar em conta que a predição final do rating incorpora não apenas os o produto interno dos vetores de fatores como também os biases (baselines) de usuário e item: $\hat{r}_{ui}=b_{ui}+p_u q_i^T=\mu+b_i+b_u+p_u q_i^T$. Biases são efeitos independentes da associação usuário-item e que, no entanto, interferem no rating final, como por exemplo "tendências" que usuários específicos têm em avaliar itens com pontuações mais altas ou baixas, ou mesmo de itens específicos de receberem pontuações especialmente baixas ou altas [7, 17, 18]. Para descobrir os valores desses parâmetros (b_u, b_i, p_u, q_i) minimizamos então o seguinte erro quadrático, tal que λ é a constante de regularização responsável por penalizar a magnitude do parâmetro em questão sendo aprendido, no fim de evitar o overfitting [18]:

$$\sum_{(u,i)\in K}(r_{ui}-\mu-b_i-b_u-p_u q_i^T)^2+\lambda(b_i^2+b_u^2+\|q_i\|^2+\|p_u\|^2)$$

Consideraremos a solução dessa minimação pelo método de otimização SGD, iterado individualmente sobre cada rating do conjunto do treinamento, resultando em uma predição ($\hat{r}_{ui}$) e um erro associado $e_{ui}=r_{ui}-\hat{r}_{ui}$. Da metodologia SGD derivam então as seguintes funções, sendo γ a taxa de aprendizado [18]:

- $b_u=b_u+\gamma(e_{ui}-\lambda b_u)$
- $b_i=b_i+\gamma(e_{ui}-\lambda b_i)$
- $q_i=q_i+\gamma(e_{ui}\,p_u-\lambda q_i)$
- $p_u=p_u+\gamma(e_{ui}\,q_i-\lambda p_u)$

Ao final das iterações SGD, tendo descoberto os parâmetros, basta aplicá-los à função $\hat{r}_{ui}$ para encontrarmos a predição, finalizando então a abordagem SVD incremental.

6 Avaliação Experimental

Este seção descreve a avaliação experimental do algoritmo SVD em relação aos algoritmos user-user e item-item, escolhidos como baselines. A escolha do algoritmo SVD deu-se por conta da sua grande popularidade, obtida após relativo sucesso na competição da Netflix Prize, tornando-a desde então uma abordagem comum, que vem adquirindo sucessivos aprimoramentos desde a competição [17, 18]. As técnicas user-user e item-item escolhidas foram em razão de já serem algoritmos tradicionalmente utilizados na literatura como baselines para avaliações com as demais técnicas de recomendação.

6.1 Conjunto de dados

Escolhemos para o experimento um conjunto de dados do MovieLens (movielens.org), um sistema de recomendação de filmes online baseado em filtragem colaborativa criado com intuitos acadêmicos pelo "GroupLens Research", em 1997, um laboratório de pesquisa do Departamento de Ciência e Engenharia da Computação da Universidade de Minnesota.

O conjunto de dados utilizado foi disponibilizado pela própria GroupLens, contendo 100 mil ratings, aplicados a 9 mil filmes e 700 usuários. Todo os usuários presentes no dataset avaliaram ao menos 20 dos filmes disponíveis e a escala de ratings compreende valores de 0 a 5, variando em múltiplos de 0.5.

6.2 Procedimento de experimentação

Utilizamos a biblioteca "Surprise", do Python, como *engine* de sistemas de recomendação para nossos experimentos e todas as técnicas foram avaliadas via experimentação com cross-validation de 10 *folds*. Utilizamos como métrica de comparação o *Root Mean Squared Error* (RMSE), amplamente utilizada na literatura de sistemas de recomendação, e o padrão de avaliação utilizado durante a competição Netflix Prize. Para ambas as técnicas de recomendação, user-user e item-item fixamos seus parâmetros com os valores que apresentaram melhor resultado:

- k: Número máximo de vizinhos – foram testados os valores “30”, “40” e “50”;
- k mínimo: Pré-requisito do número mínimo de vizinhos existentes, caso contrário a predição toma o valor da média global dos ratings – foram testados os valores “1”, “5” e “10”;
- Função-similaridade: Função utilizada no cálculo da similaridade entre vetores de ratings – a função cosseno similaridade foi tomada como base em ambos, user-user e item-item.

Em ambos os casos utilizamos no cálculo da predição a função de normalização de centralização pela média, discutido no tópico 4.1. Executados os algoritmos, a melhor permutação de parâmetros encontrada para o algoritmo user-user teve valor k de “50” vizinhos e k mínimo “5”; o RMSE encontrado foi de “0,911709”. Para o algoritmo item-item a melhor permutação encontrada teve valor k de “50” vizinhos e k mínimo “1”; o RMSE encontrado foi de “0,91778”.

Para a técnica SVD variamos os parâmetros de “Número de fatores”, “Taxa de aprendizado” e “Termo de regularização”:

- Número de fatores: Testamos os valores “100”, “200”, “300”, “400” e “600”;
- Taxa de aprendizado: Testamos os valores “0,005”, “0,01” e “0,05”;
- Termo de regularização: Testamos os valores “0,05”, “0,1” e “0,5”.

6.3 Resultados

Para cada valor de parâmetro testado no SVD computamos uma média dos RMSE’s obtidos e montamos gráficos desses resultados abaixo:

Figura 1 – Taxa de Aprendizado x RMSE

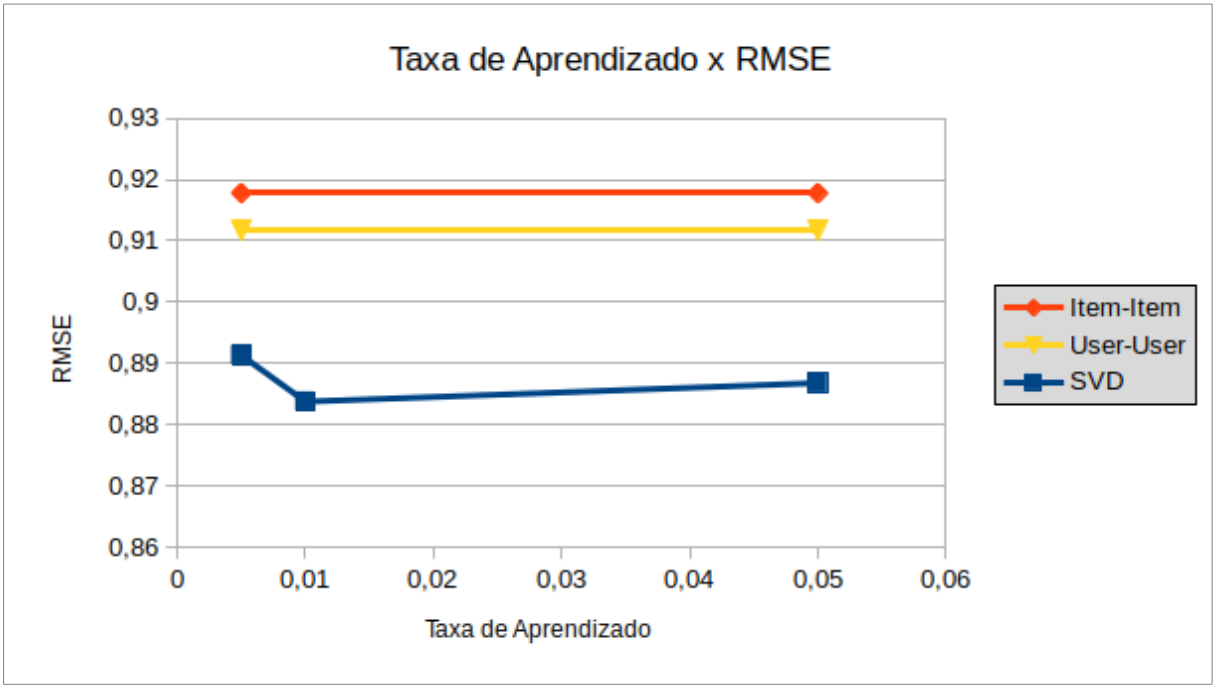


Figura 2 – Termo de Regularização x RMSE

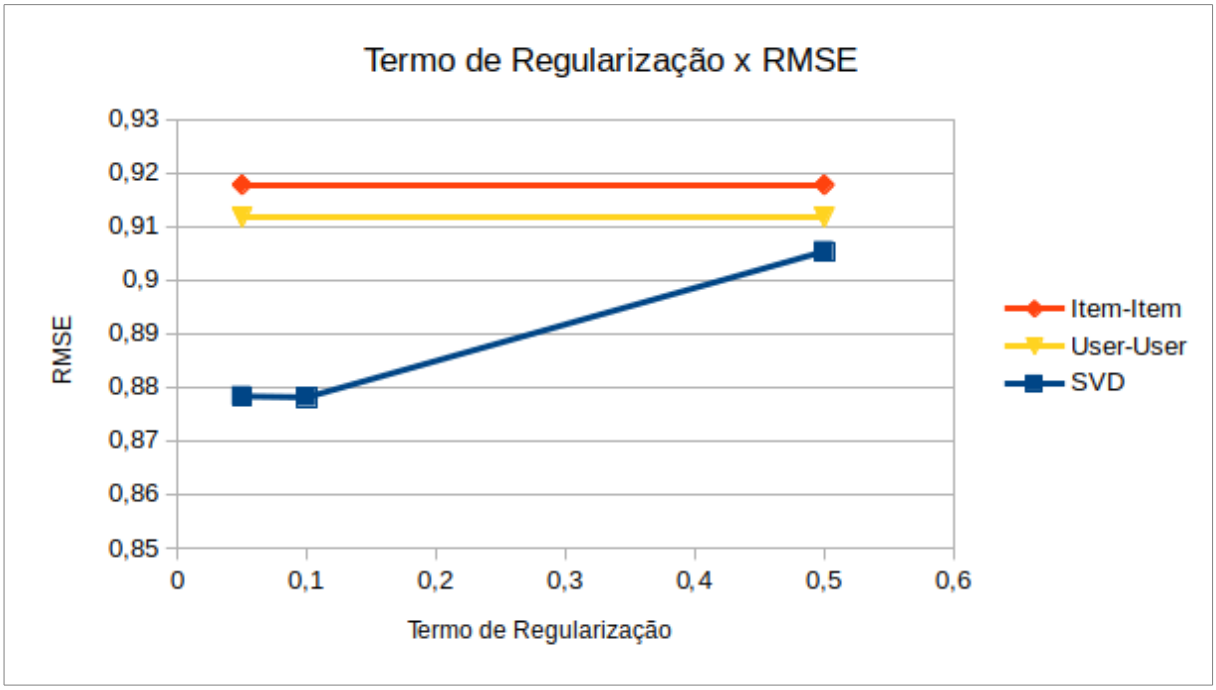


Figura 3 – Quantidade de features x RMSE (comparativo SVD, user-user e item-item)

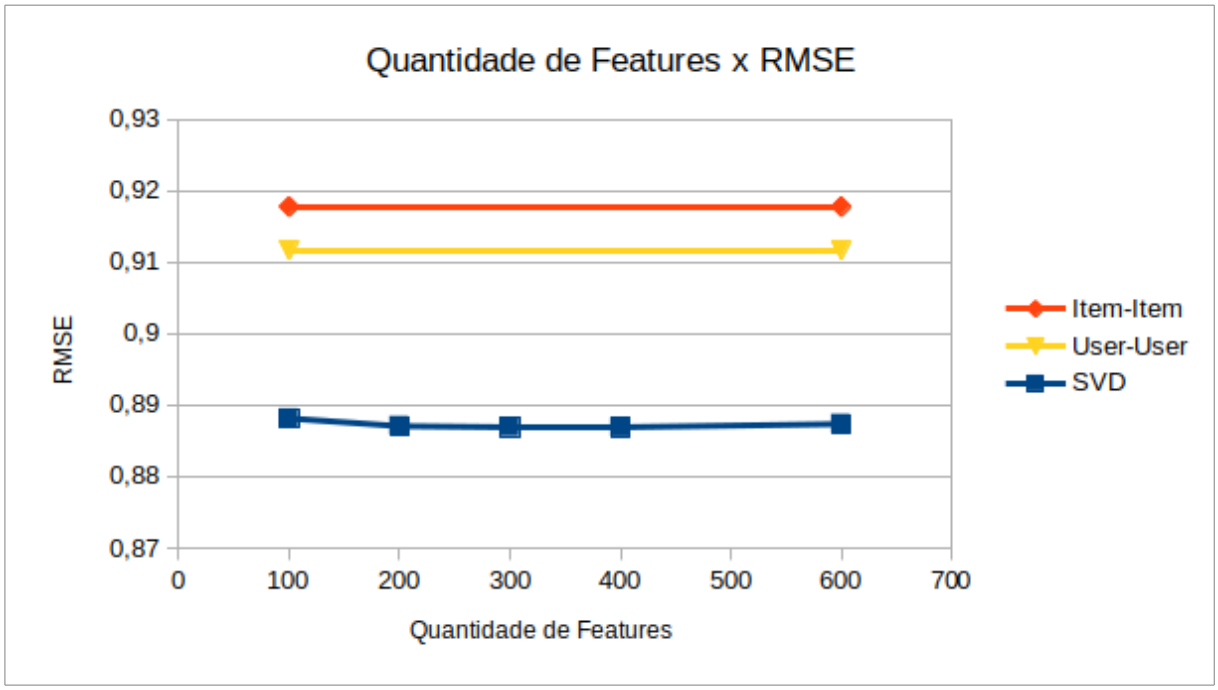
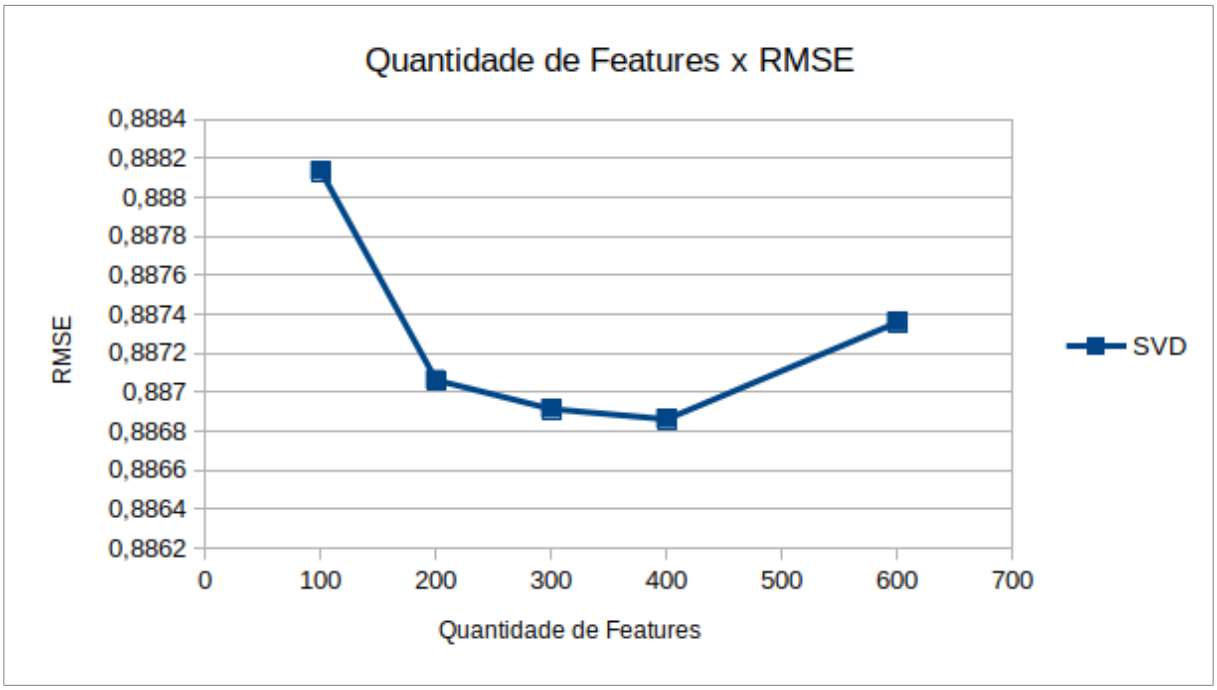


Figura 4 – Quantidade de features x RMSE



Analisando os resultados dos gráficos mostrados, percebe-se que o SVD mostrou-se mais preciso que as técnicas item-item e user-user em geral, e que obteve bons resultados mesmo com uma quantidade de features/fatores reduzida,. Isso significa um melhor custo-benefício em comparação com as demais técnicas, dado que é possível obter via SVD uma melhor precisão que nas demais, sendo necessário menor espaço para armazenar o modelo gerado pela técnica, por poder gerar seu modelo com poucas features em comparação à matriz de similaridades necessária às técnicas user-user e item-item. A melhor generalização, principalmente decorrente da possibilidade de itens similares por transitividade e reconhecimento de sinônimos pode ter sido um fator essencial nos resultados obtidos com o SVD.

7 Conclusão e Trabalhos Futuros

Abordamos o conceito de sistemas de recomendação, suas ramificações e vantagens gerais, partimos então mais detalhadamente ao contexto de técnicas de filtragem colaborativa. Vimos que são as técnicas mais comumente utilizadas em sistemas de e-commerce e que a competição "Netflix Prize" foi responsável por um grande salto nos avanços dessa área, estimulando entusiastas do campo a retornarem seu foco aos sistemas de recomendação e, principalmente, à área de filtragem colaborativa.

Explicamos o funcionamento das três técnicas de filtragem colaborativa – user-user, item-item e SVD – que foram usadas em nossos experimentos e as vantagens da redução de dimensionalidade. Falamos do conjunto de dados utilizado nos experimentos (MovieLens), da engine de recomendação utilizada (Surprise) e o procedimento utilizado durante as experimentações.

Vimos então que os resultados obtidos para o nosso dataset indicaram uma clara vantagem da técnica SVD em relação ao user-user e item-item, independente dos valores dados aos parâmetros de configuração do SVD, vimos inclusive que seria possível reduzir os custos de espaço do sistema de recomendação ao diminuir o parâmetro "Quantidade de features" do SVD, dada a diferença mńima nos resultados de RMSE quando comparadas a valores mais altos. Sabemos, no entanto, a dificuldade do SVD em explicar as recomendações oferecidas ao usuário, dada a camada intermediária de fatores latentes gerada, se comparada a recomendações derivadas da técnica item-item.

Em trabalhos futuros poderiam ser testados novos datasets para contrapor os resultados já obtidos a fim de se observar melhor os impactos do parâmetro de "quantidade de features" e sinais de overfitting, além de utilizar outras técnicas de SVD incremental e model-based em geral. Uma possível boa alternativa seria o SVD++, uma derivação do SVD que faz também uso de feedbacks implícitos, utilizada durante a Netflix Prize com relativo sucesso, assim como abordagens híbridas das alternativas memory-based e model-based.

Referências Bibliográficas

[1] SCHAFER, J. Ben; KONSTAN, Joseph; RIEDL, John. **Recommender Systems in E-Commerce**. University of Minnesota, Minneapolis: Department of Computer Science and Engineering, GroupLens Research Group; 1999.

[2] DAVIDSON, James et al. **The YouTube Video Recommendation System**. 2010.

[3] LESKOVEC, Jure; RAJARAMAN, Anand; ULLMAN, Jeff. **Mining of Massive Datasets**. 2ed; 2014. Disponível em: http://www.mmds.org/#ver21.

[4] BEEL, Joeran et al. **Research Paper Recommender System Evaluation: A Quantitative Literature Survey**. 2013.

[5] SARWAR, Badrul; KARYPIS, George; KONSTAN, Joseph, et al. **Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems**. University of Minnesota, Minneapolis: Department of Computer Science and Engineering, GroupLens Research Group; 2002.

[6] LINDEN, Greg; SMITH, Brent; YORK, Jeremy. **Amazon.com Recommendations: Item-to-Item Collaborative Filtering**. IEEE Computer Society. Industry Report; 2003.

[7] KOREN, Yehuda. **Factor in the Neighbors: Scalable and Accurate Collaborative Filtering**. ACM Transactions on Knowledge Discovery from Data: Yahoo! Research; 2010.

[8] WITTEN, Ian H.; FRANK, Eibe; HALL, A. Mark. **Data Mining: Practical Machine Learning Tools and Techniques**. 3ed; 2011.

[9] SARWAR, Badrul M; KARYPIS, George; KONSTAN, Joseph A, et al. **Application of Dimensionality Reduction in Recommender System -- A Case Study**. University of Minnesota, Minneapolis: Department of Computer Science and Engineering, GroupLens Research Group; 2000.

[10] SU, Xiaoyuan; KHOSHOFTAAR, M. Taghi. **A Survey of Collaborative Filtering Techniques**. Florida Atlantic University: Department of Computer Science and Engineering; 2009.

[11] PORIYA, Anil et all. **Non-Personalized Recommeder Systems and User-based Collaborative Recommender Systems**. International Journal of Applied Information Systems (IJAIS). Foundation of Computer Science FCS, New York, USA. Volume 6 – No. 9, March 2014.

[12] JONES, M. Tim. **Recommender systems, Part 1: Introduction to approaches and algorithms**. IBM: developerWorks; 2013.

[13] DOMINGOS, Pedro. **A Few Useful Things to Know about Machine Learning**. University of Washington, Seattle: Department of Computer Science and Engineering; 2012.

[14] KONSTAN, Joseph ; RIEDL, John. **Desconstructing Recommender Systems – How Amazon and Netflix predict your preferences and prod you to purchase**. 2012. Disponivel em: http://spectrum.ieee.org/computing/software/deconstructing-recommender-systems

[15] LEE, Joonseok; SUN, Mingxuan; LEBANON, Guy. **A Comparative Study of Collaborative Filtering Algorithms**. 2012.

[16] LEE, Jong Seo. **Survey of Recommender Systems (Collaborative Filtering)**. California Polytechnic State University.

[17] GOWER, Stephen. **Netflix Prize and SVD**. 2014, April 18th.

[18] RICCI, Francesco; ROKACH, Lior; SHAPIRA, Bracha, et al. **Recomender Systems Handbook**. 1ed. Editora Springer US, 2011.