



Trabalhos de Conclusão de Curso
13 de novembro, São Paulo-SP, Brasil

DESENVOLVIMENTO DE UM SISTEMA DE RECOMENDAÇÃO PARA E-COMMERCE

Antonio Guilherme Ferreira Viggiano
agfviggiano@gmail.com

Fernando Fochi Silveira de Araújo
fernando.fochi@gmail.com

Orientador: Prof. Dr. Fábio Gagliardi Cozman
fgcozman@usp.br

Resumo

O objetivo deste trabalho é o desenvolvimento de um Sistema de Recomendação de produtos para e-Commerces e respectiva análise de desempenho das recomendações. Serão propostos três diferentes algoritmos de recomendação, e será feita uma avaliação comparativa entre eles. No nosso trabalho, o sistema terá como foco a sugestão de produtos de lojas de comércio online que disponham de um histórico de compras dos usuários e das características dos produtos. A qualidade das recomendações será avaliada quanto a precisão e abrangência. Por meio de uma validação cruzada, analisaremos a influência dos parâmetros do problema na qualidade das recomendações, como o tamanho do banco de dados ou a quantidade de informações utilizada de itens e clientes. Serão discutidos os principais desafios tecnológicos e científicos dos sistemas de recomendação quanto a nossa proposta de solução, tais como a escalabilidade, a adaptação a novos usuários e a esparsidade dos dados. Ao final, será possível extrair uma validação experimental das diretrizes fundamentais a serem seguidas por comércios on-line que desejem criar um sistema de recomendação próprio ou que queiram utilizar o sistema desenvolvido neste trabalho.

Palavras chave: Sistema de Recomendação, e-Commerce, Validação cruzada, Recomendação híbrida.

1. Introdução

O comércio online se torna cada vez mais importante na vida das pessoas, de forma que a adoção deste método de compra é cada vez mais comum. Estima-se que em 2013 um bilhão de pessoas compraram online (1), gerando uma

receita anual de 1,25 trilhão de dólares com expectativa de crescimento de 17% ao ano até 2017 (2). Como o mercado de varejo online é novo como um todo, este ainda passará por algumas mudanças drásticas em um curto espaço de tempo. Um dos itens-chave destas mudanças é a capacidade de se analisar os dados gerados pelos consumidores. Com estes dados será possível segmentar os clientes mais facilmente, dando possibilidade às empresas de direcionar suas estratégias de forma mais eficiente, até o ponto em que campanhas de marketing e precificação serão totalmente personalizadas (3). Uma das maneiras de se usar estes dados é através dos sistemas de recomendação. Sistemas de recomendação são “ferramentas e técnicas de software destinadas a prover sugestões de itens para usuários” (4). Estes sistemas são utilizados por diversos serviços online e geram um grande impacto quando utilizados corretamente. Em 2012, cerca de 75% dos vídeos assistidos através do site Netflix foram acessados por meio de recomendações (5). Em 2006, as recomendações representaram 35% dos livros vendidos pela Amazon (6), enquanto em 2007 cerca de 38% das notícias lidas no Google News foram sugeridas por um sistema de recomendação (7). Para atingir uma alta qualidade de sugestões, é imprescindível que o sistema seja altamente personalizado. Essa ideia se resume ao comentário de um antigo diretor executivo do e-commerce Amazon.com, de que “se você possui 2 milhões de clientes na web, você precisa ter 2 milhões de lojas na web”.

2. Metodologia

Por se tratar primordialmente de um projeto de Engenharia de Software, a metodologia de projeto deste



trabalho é fundamentalmente iterativa. A primeira etapa, de definição de necessidade, decorreu da constatação de que o crescente número de lojas de comércio online impulsiona a criação de sistemas de recomendação gratuitos, de fácil integração e de código aberto. Em seguida, quanto à definição dos parâmetros de sucesso, estabelecemos que deveria ser feita uma verificação entre as sugestões do sistema e os itens comprados pelos usuários. Posteriormente, foram propostas diversas soluções para o problema. O processo de escolha levou em conta dois pontos: a viabilidade da solução, principalmente ditado pela reprodutibilidade dos algoritmos apresentados nos artigos científicos, e a sua adequação aos requisitos de projeto e aos bancos de dados de e-commerces. Foram escolhidos dois métodos-base, provenientes de artigos de referência, e um terceiro método proposto pela dupla. Após o primeiro semestre de trabalho, os algoritmos foram implementados e testados em bancos de dados simples, com menos de dez usuários e itens.

Após essa primeira etapa de projeto, passamos à fase de aplicação das soluções em bancos de dados reais. Para este semestre, utilizaremos bancos de dados de cem mil compras, e faremos uma validação cruzada a fim de avaliar a qualidade de cada um dos métodos. Assim que a fase de aplicação for concluída com êxito, serão feitas melhorias incrementais no projeto, a fim de deixá-lo mais genérico.

3. Requisitos do Projeto

Os requisitos funcionais de um sistema de recomendação ditam principalmente sobre a escalabilidade e o desempenho das recomendações do sistema. Como as sugestões serão calculadas de maneira *offline*, não há necessidade para uma elevada taxa de recomendações por período de tempo (*throughput*). Deseja-se contudo que o sistema possa gerar todas as recomendações para um banco de dados de cem mil clientes em uma hora, isto é, que tenha *throughput* mínimo de 28 recomendação por segundo. Os sistemas de recomendação tradicionais possuem *throughput* de cerca de 500 recomendações por segundo, mas operam em servidores dedicados de maior potência computacional (8). A fim de poder estabelecer uma base comparativa entre o sistema proposto pela dupla UI e os sistemas de referência FW e UP, serão utilizados os mesmos indicadores de desempenho dos artigos-base: precisão, abrangência e medida F_1 (9, 10). Precisão é a porcentagem de casos corretamente preditos em relação ao tamanho da lista de recomendações. Abrangência é a razão entre o número de itens corretamente preditos e aqueles que foram efetivamente avaliados pelo usuário. A medida F_1 , por sua vez, é a média harmônica entre precisão e abrangência. Todas

essas métricas são dependentes dos diversos parâmetros do problema, como do tamanho da lista de recomendações N , da quantidade de vizinhos mais próximos k , e principalmente do banco de dados utilizado. Como os artigos de referência não o disponibilizaram integralmente, os valores de precisão, abrangência e medida F_1 serão estimados para o banco de dados da dupla. Espera-se que a precisão, abrangência e consequentemente a medida F_1 sejam maiores que 20%. Esses valores foram escolhidos por serem superiores aos de algoritmos puramente baseados em conteúdo ou em filtragem colaborativa (9, 10). Na prática, o resultado mais importante é a comparação entre os três métodos para um mesmo banco de dados de referência.

Tabela 1: Requisitos para o sistema de recomendação

Requisito	Valor
Throughput	28 recomendações/segundo
Precisão	20 %
Abrangência	20 %

4. Síntese de Soluções

4.1. Algoritmo baseado na ponderação de atributos (FW)

O primeiro algoritmo que utilizaremos no sistema de recomendação, adaptado de (9) e denominado ponderação de atributos, *feature weighting* ou FW, trata-se de um híbrido entre filtragem colaborativa e filtragem baseada em conteúdo. A partir da regressão linear de dados de uma rede social (*Internet Movie Database, IMDB*), extraem-se os pesos que determinam a importância de cada atributo dos itens, onde ocorre a filtragem colaborativa dos usuários. Após obtenção dos pesos, realiza-se a filtragem baseada em conteúdo para determinar os itens com maior similaridade, que são finalmente recomendados. Na filtragem baseada em conteúdo, “cada item é representado por um vetor de atributos ou features”. A similaridade s_{ij} entre dois itens i e j é dada pela média ponderada das distâncias entre as *features* dos itens:

$$s_{ij} = \sum_f w_f (1 - d_{ij}^f)$$

As distâncias entre os atributos d_{ij}^f são determinadas conforme o tipo de dado avaliado e seu domínio, normalizadas no intervalo $[0, 1]$. Os pesos w_f são a priori desconhecidos. A Referência (9) os determina a partir de uma regressão linear, onde se determina o número de usuários que se interessam tanto por i quanto por j . Esses valores permitem determinar “o julgamento humano de similaridade entre itens”, e pode ser calculado a partir da matriz de avaliações. Desta forma, os pesos w_f são determinados e calcula-se a matriz de similaridade S . Recomendam-se, por fim, os itens similares àqueles já comprados.



4.2. Algoritmo baseado no perfil de usuários (UP)

O segundo algoritmo, adaptado de (10), é um híbrido entre filtragem colaborativa e filtragem baseada em conteúdo. Os atributos dos itens são ponderados no cálculo de similaridade, com pesos extraídos de um modelo de perfil de usuários, denominado *user profile* ou UP. Esse perfil leva em consideração o interesse dos usuários por features, indiretamente calculado a partir de seu interesse pelos itens. Para se determinar a relevância de f para u , deve-se levar em conta não somente a frequência com a qual uma característica aparece, mas também o fato de algumas características estarem contidas na maioria dos itens. Determina-se, então, os pesos w_{uf} , que mostram a relevância de f para u , a partir da medida estatística TF-IDF (*term frequency-inverse document frequency*), presente em formulações de recuperação de informação e mineração de dados.

$$w_{uf} = \text{TF}_{uf} \text{IDF}_f$$

Os pesos w_{uf} , obtidos na TF-IDF, são utilizados para calcular a similaridade s_{uv} entre dois usuários u e v . Dispondo-se de S , selecionam-se os k vizinhos mais próximos v_u^k com maior similaridade s_{uv} , dentre todos $v \neq u$. Posteriormente, determina-se o conjunto Iv_u^k de itens i avaliados positivamente por v_u^k . Avalia-se a frequência total f_{uf} dos atributos f para os itens de Iv_u^k . Por fim, calcula-se o peso ω_{ui} de cada item, a partir da matriz de avaliações a_{if} e de frequências f_{uf} , e gera-se a lista dos top- N produtos a serem recomendados para o usuário u .

$$\omega_{ui} = \sum_f a_{if} f_{uf}$$

4.3. Algoritmo baseado na correlação usuário-item (UI)

Este método se trata de uma variante da solução UP, e também está embasado no cálculo da preferência do usuário por features, medida através do seu interesse pelos itens. O algoritmo UI utiliza as matrizes de correlação ponderada entre usuários e atributos W e a matriz de atributos dos itens A no cálculo da correlação usuário-item. A lista dos N produtos a serem recomendados decorre portanto do cálculo de ω_{ui} e da escolha dos itens que maximizem essa variável para cada usuário.

$$\omega_{ui} = \sum_f w_{uf} a_{if}$$

Ao passo que o método UP recomenda itens a partir dos k vizinhos mais próximos, o algoritmo UI busca os itens com features mais similares aos atributos pelos quais u se interessa, diretamente através da matriz de atributos. Espera-se que esse tipo de recomendação forneça sugestões de qualidade similar ao algoritmo original, pois os dois tem a mesma fundamentação.

5. Resultados

Os resultados da primeira etapa deste Trabalho de Conclusão de Curso, realizadas na disciplina PMR2500, foram principalmente a definição de necessidades, de parâmetros de sucesso e a elaboração de possíveis soluções. Definimos que a aquisição de dados seria feita a partir de uma base que deveria alimentar o sistema por meio de arquivos de texto com valores separados por vírgulas (.csv). A fim de facilitar o pré-processamento dos dados, estabelecemos que seriam necessários dois arquivos. Um deles deve conter a matriz de atributos A e o outro, a matriz de avaliações R . Uma vez determinada a forma de entrada de informações, definiram-se os conjuntos de dados que serão utilizados. O primeiro conjunto de dados abertos é proveniente do sistema de recomendações de filmes MovieLens (<http://movielens.umn.edu>). Nessa base de dados, o catálogo de filme faz o papel de catálogo de produtos, e o histórico de compras se refere à avaliação dos filmes feita por cada usuário. Outro conjunto de dados abertos é do website Internet Movie Database (IMDB). Na nossa análise, esses dois bancos poderão ser utilizados complementar ou independentemente. Na primeira etapa do projeto, buscamos parcerias com e-commerces que estivessem dispostos a doar anonimamente seu banco de dados. Há ainda a possibilidade de utilizarmos uma terceira base, mas visto que as negociações ainda não foram concluídas, daremos prioridades aos conjuntos *open source*.

A partir da síntese de soluções estabelecida na primeira etapa do projeto implementamos os três algoritmos de recomendação e as medidas de recomendação na linguagem de programação estatística R. O código já está disponível para consulta através do endereço <https://github.com/aviggiano/tcc>. Após a etapa de implementação, fizemos o tratamento das bases de dados, adequando-as ao formato de entrada especificado, e iniciamos o processo de desenvolvimento do *cross-validation*.

5.1. Tamanho da lista de recomendações

A medida que o tamanho da lista de recomendações aumenta, a precisão cai e a abrangência cresce (Figuras 1 e 2). A primeira decresce com N porque a quantidade de itens sugeridos se torna excessivamente maior que a quantidade de itens positivamente avaliados pelos usuários-teste. A segunda, por sua vez, cresce com N porque a probabilidade de sugerirmos itens relevantes para o usuário aumenta quando sugerimos mais itens. Para $N = |I|$ a abrangência atinge 100%, pois todos os itens teriam sido recomendados.

O método UP supera os outros algoritmos para todos os valores de N , tanto em precisão quanto em abrangência, como se observa pelo gráfico das medidas F_1 .



4

Figura 1 - Precisão por tamanho de N

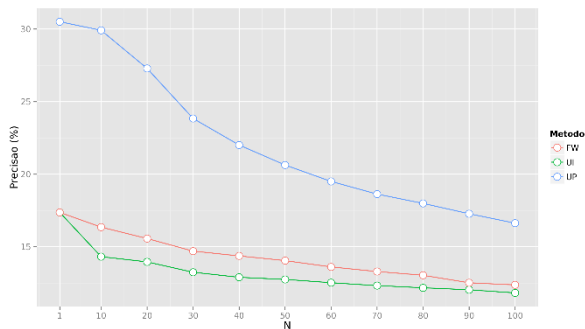


Figura 2 - Abrangência por tamanho de N

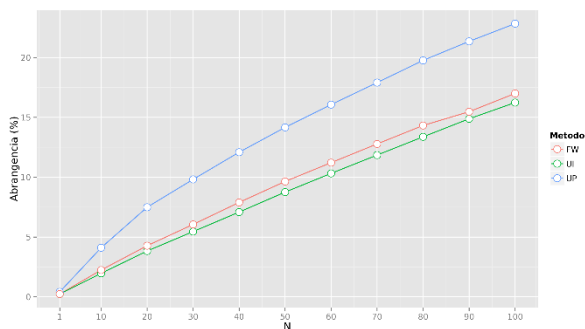
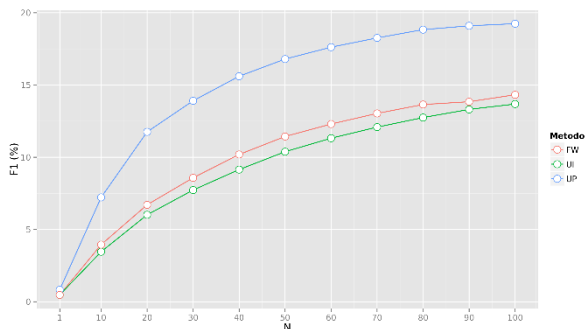
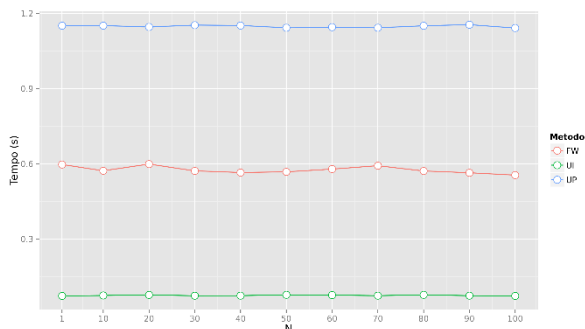


Figura 3 - Medida F_1 por tamanho de N



Apesar de alta qualidade das recomendações do método UP, este possui também a maior complexidade computacional. Seu tempo de execução é 2 vezes maior que o do método FW e 4 vezes maior que o do método UI, como podemos ver na figura 4.

Figura 4 - Tempo de execução por tamanho de N



5.2. Percentual da base de aprendizado T

A medida que o percentual da base de aprendizados aumenta, a precisão de todos os métodos cresce ligeiramente. Isso é consequência do caráter colaborativo dos algoritmos, já que a qualidade da recomendação depende da quantidade total de dados. Entretanto, pode-se observar que a abrangência e a medida F_1 são praticamente constantes para valores crescentes de T de modo que esse parâmetro não tem grande relevância para o sucesso do sistema de recomendação.

O parâmetro T não exerce nenhuma influência sobre o tempo de execução dos métodos UP e UI, mas apenas sobre o método FW. Isso ocorre porque a etapa de maior custo computacional é linearmente dependente da quantidade de usuários $|U|$. Quanto menos usuários-teste, mais veloz é o algoritmo.

Figura 5 - Precisão por percentual da base de aprendizado T

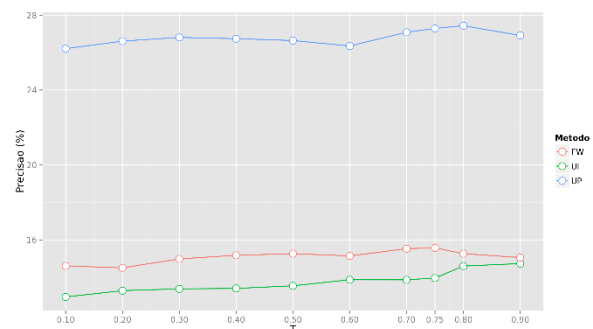


Figura 6 - Abrangência por percentual da base de aprendizado T

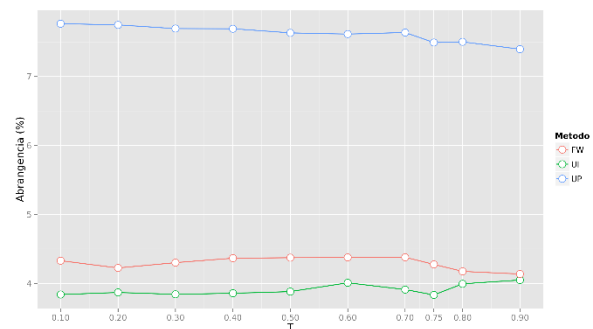
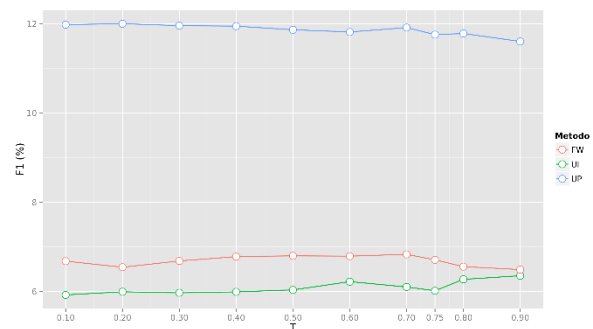


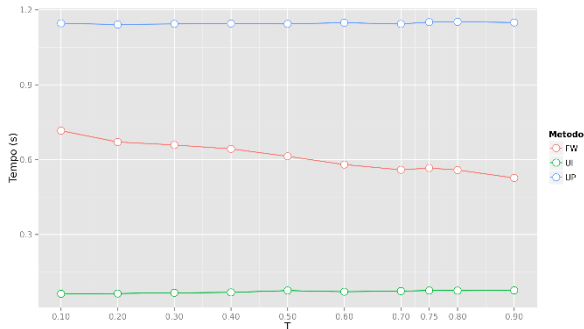
Figura 7 - Medida F_1 por percentual da base de aprendizado T





5

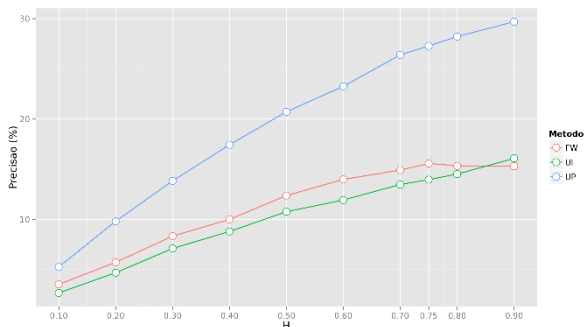
Figura 8 - Tempo de execução por percentual da base de aprendizado T



5.3. Percentual de avaliações “escondidas” dos usuários-teste na validação cruzada H

Quanto maior o número de avaliações “escondidas”, mais fácil é acertar os itens dos usuários-teste, pois a lista de recomendação é pequena em relação ao total de itens positivamente avaliados pelo usuário. Por esse motivo, a precisão cresce com H para todos os métodos. Para o algoritmo FW, a precisão atinge seu máximo em $H = 75\%$ e depois decresce ligeiramente (Figura 9).

Figura 9 - Precisão por percentual de avaliações escondidas H



Isso ocorre porque o cálculo dos pesos w_f depende da quantidade de avaliações r_{ui} . Existe, pois, um compromisso (tradeoff) entre facilidade de se acertar itens avaliados quando há muitas avaliações escondidas e a dificuldade de se estimar w_f quando não há muitos dados de avaliações.

Ao passo que a precisão dos métodos aumenta com H , a abrangência diminui. Visto que a quantidade de itens da lista $top-N$ é fixa, quanto maior o número de itens “escondidos”, mais difícil é de se retornar todos os itens relevantes.

O resultado de uma precisão crescente em função de H e uma abrangência decrescente é que a medida F_1 possui um ponto de máximo. Para todos os métodos, o valor máximo é tal que $H = 50\%$ (Figura 11).

Quanto ao tempo de execução, a influência é a mesma do parâmetro T : para o método FW, a etapa de maior custo computacional é linearmente dependente da quantidade de itens $|I|$. Quanto menos avaliações de itens dos

usuários-teste, mais veloz é o algoritmo.

Figura 10 - Abrangência por percentual de avaliações escondidas H

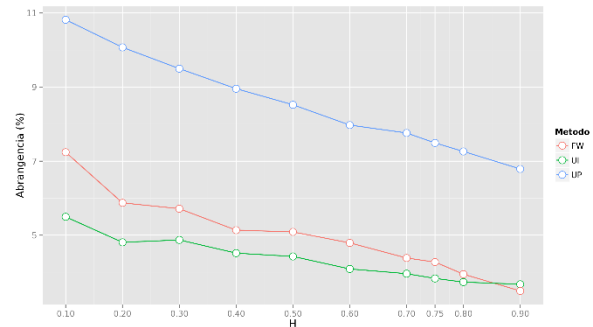


Figura 11 - Medida F_1 por percentual de avaliações escondidas H

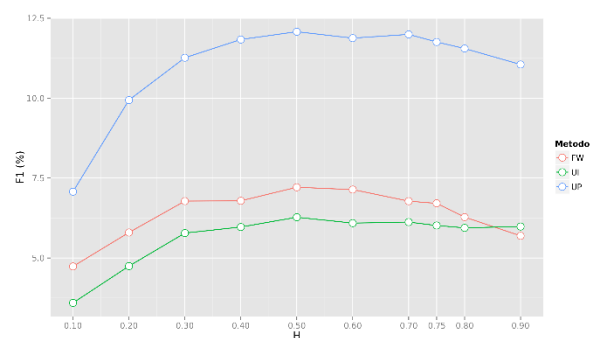
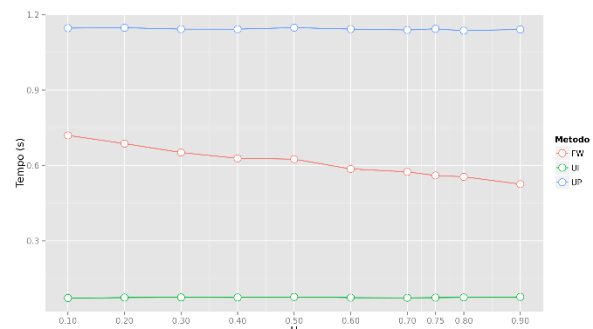


Figura 12 - Tempo de execução por percentual de avaliações escondidas H



5.4. Valor mínimo para avaliações positivas M

Contrariamente ao que esperávamos, tornar o algoritmo mais “seletivo” não melhora sua precisão. Apesar de o valor mínimo M estar intimamente ligado com a noção de “avaliação positiva” e de entrar no cálculo de parâmetros importantes dos métodos, esse parâmetro pouco influencia a precisão para $0 \leq M \leq 2$.

Esse resultado pode ser explicado porque a maioria das avaliações são positivas (Figura 14), e portanto b_M tem quase o mesmo efeito de b_0 . Isso não ocorre somente pelo fato de os clientes comprarem itens similares a seus gostos, e portanto de raramente se decepcionarem, mas também pelo fato de os usuários terem menos disposição



6

para dar avaliações negativas. Esse fenômeno se chama *hidden feedback*, e se caracteriza pelo fato de que os itens avaliados não são escolhidos ao acaso, mas sim por despertarem aspectos de interesse das preferências do usuário, indo além dos valores numéricos das avaliações (12).

Figura 13 – Precisão por valor mínimo para avaliações positivas M

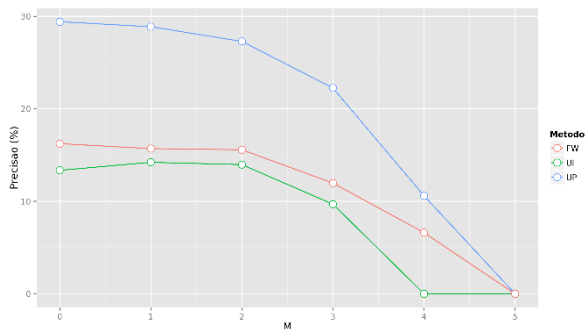
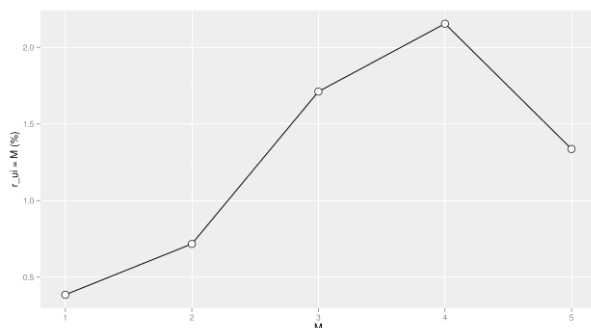


Figura 14 - Percentual de avaliações por valor de M



Ao se analisar a abrangência dos métodos, a seletividade influencia na recomendação. Quanto maior M , menor é a quantidade de itens muito bem avaliados. Estes possuem elevada ponderação/correlação e são facilmente escolhidos pelos algoritmos. Por esse motivo, o desempenho do sistema é melhor.

A complexidade computacional dos algoritmos também depende de M , já que mais ou menos parâmetros são analisados no cálculo da TF-IDF (métodos UI e UP) e dos pesos dos atributos (método FW).

Um detalhe a se observar é que a precisão é nula e a abrangência é inexistente para $M = 5$, já que todas as avaliações r_{ui} pertencem ao conjunto $\{1, 2, 3, 4, 5\}$. Para o algoritmo UI, tanto a precisão quanto a abrangência são nulas para $M = 4$.

Figura 15 - Abrangência por valor mínimo para avaliações positivas M

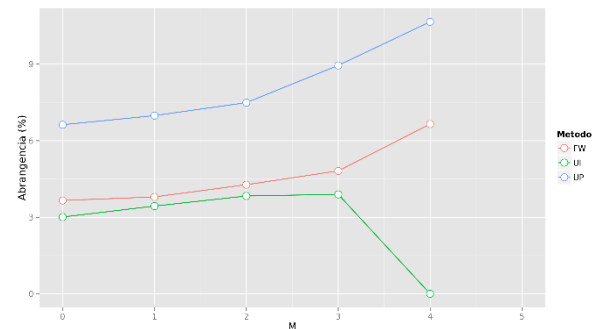


Figura 16 – Medida F_1 por valor mínimo para avaliações positivas M

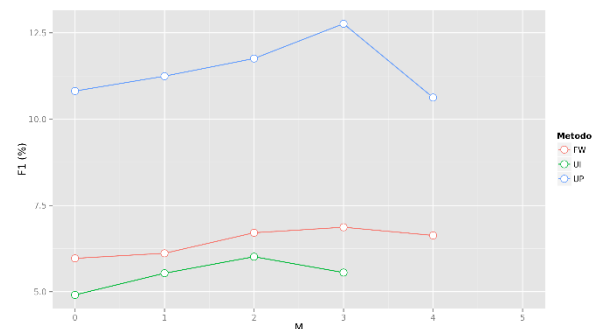
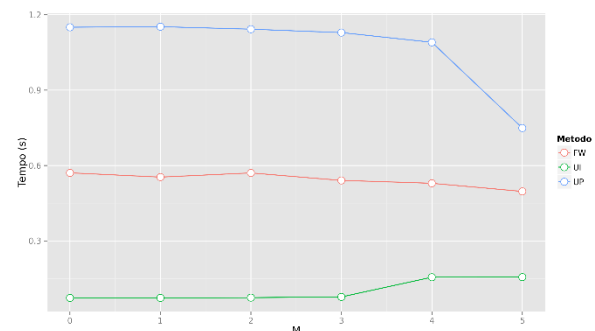


Figura 17 – Tempo de execução por valor mínimo para avaliações positivas M



5.5. Número de vizinhos mais próximos k

O único método que recomenda itens com base nos vizinhos mais próximos é o UP. Percebe-se que com o aumento de k , a precisão e a abrangência caem, pois a vizinhança se torna excessivamente grande e repleta de usuários sem muita similaridade com o usuário-teste. Pode-se observar que o valor máximo de precisão e abrangência ocorre para $k = 20$ (Figura 20).



Figura 18 - Precisão por número de vizinhos mais próximos k

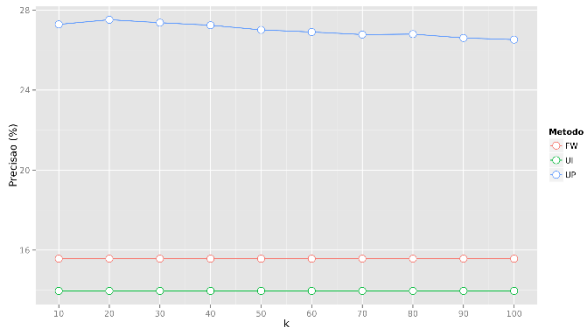


Figura 19 - Abrangência por número de vizinhos mais próximos k

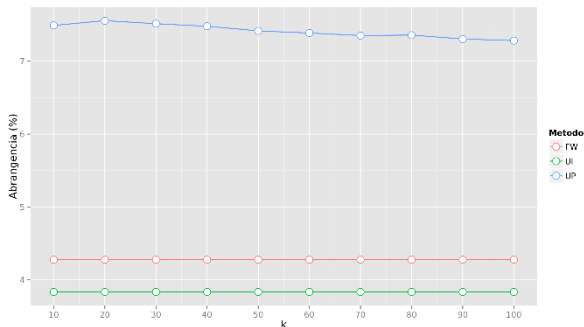
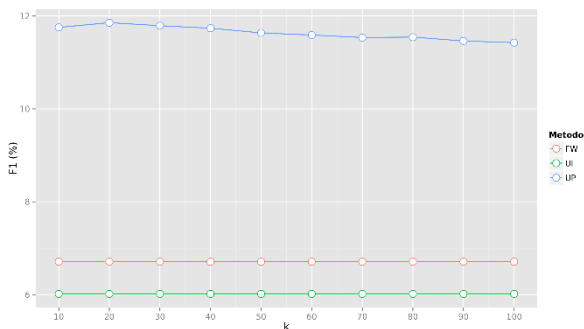


Figura 20 - Medida F_1 por número de vizinhos mais próximos k



5.6. Conjunto de atributos dos itens F

Para o banco de dados 100k-IMDB, o conjunto de atributos dos itens é $F = \{\text{data de lançamento, gênero, ano, duração, orçamento, avaliação, votos}\}$. A fim de se avaliar a performance dos algoritmos mediante a remoção em determinados atributos, decidimos excluir do conjunto as features $\{\text{data de lançamento, ano}\}$, pois julgamos que elas não eram tratadas corretamente pelos métodos UI e UP.

O resultado desse experimento se observa por exemplo na Figura 21, em que a precisão de todos os métodos melhora substancialmente. Da mesma forma, a abrangência também aumenta para todos os métodos, como se vê na Figura 22.

A conclusão desse experimento é que aumentar a

quantidade de atributos dos itens não aumenta necessariamente a qualidade do algoritmo de recomendação. De fato, o algoritmo deve estar preparado para “aprender” quais são as features relevantes para cada usuário e eventualmente descartar automaticamente os atributos desnecessários.

Figura 21 - Precisão por tamanho da lista de recomendações N excluindo-se alguns atributos f

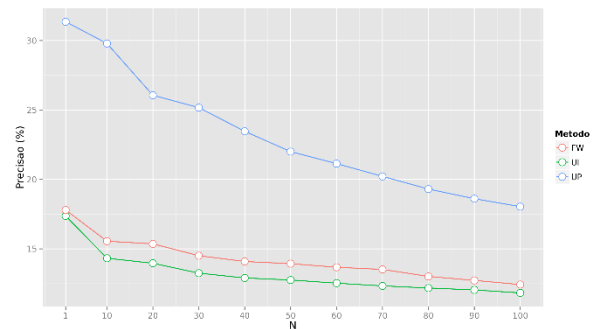


Figura 22 - Abrangência por tamanho da lista de recomendações N excluindo-se alguns atributos f

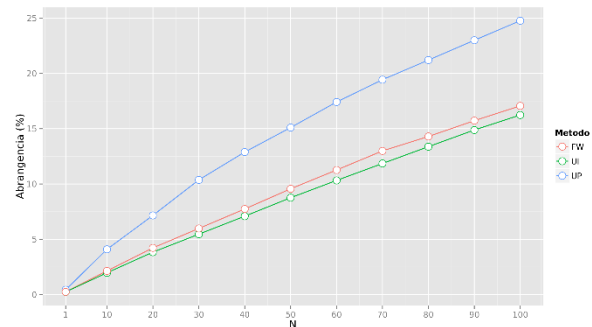
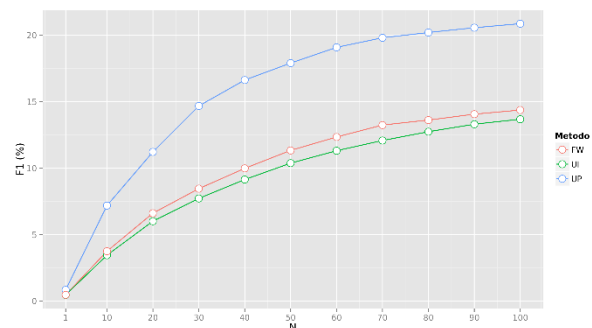


Figura 23 - Medida F_1 por tamanho da lista de recomendações N excluindo-se alguns atributos f



5.7. Medida de distância entre atributos d^f

Contrariamente ao esperado, o emprego de medidas de distância específicas para o método FW não melhorou a qualidade da recomendação. O uso do índice de Jaccard para os atributos do tipo gênero, em vez da distância absoluta, decresceu a precisão e abrangência do método.



8

Nota-se a importância da determinação de uma medida de distância correta, pois a influência no desempenho do método é drástica.

Figura 24 - Precisão por tamanho da lista de recomendações N para diferentes medidas d^f

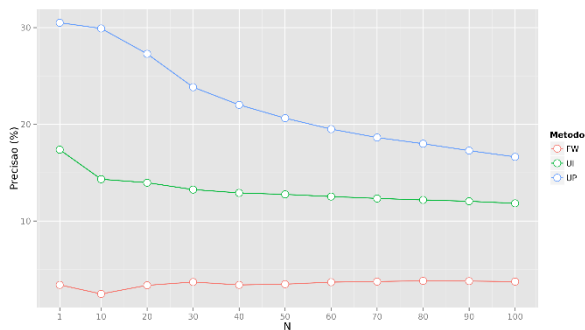


Figura 25 - Abrangência por tamanho da lista de recomendações N para diferentes medidas d^f

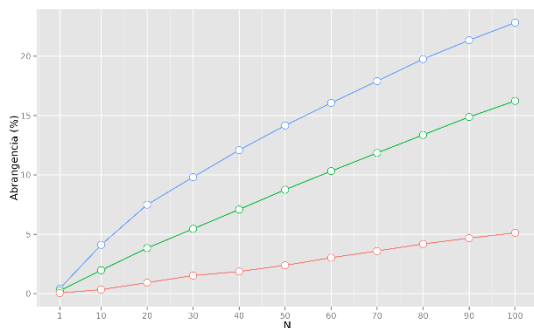
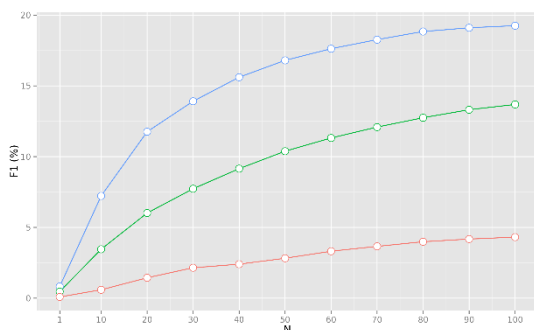


Figura 26 - Medida F_1 por tamanho da lista de recomendações N para diferentes medidas d^f



5.8. Pesos dos atributos w_f

A fim de avaliar a influência da quantidade de pesos utilizados na recomendação do método FW, realizamos os testes selecionando apenas os W maiores elementos $w_f > 0$.

Observa-se que a qualidade da recomendação, tanto em termos de precisão, abrangência e tempo de execução, são aproximadamente independentes da quantidade de pesos W . Para $W > 12$, as medidas de desempenho se

tornam constantes.

Figura 27 - Precisão por quantidade de pesos W

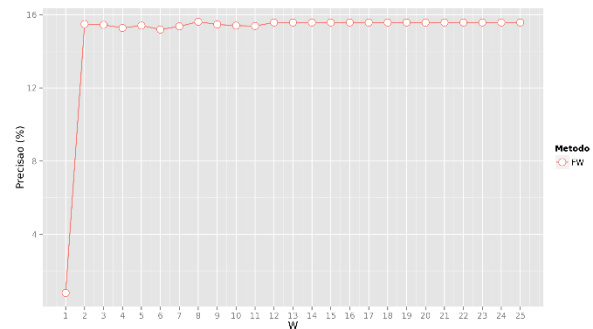


Figura 28 - Abrangência por quantidade de pesos W

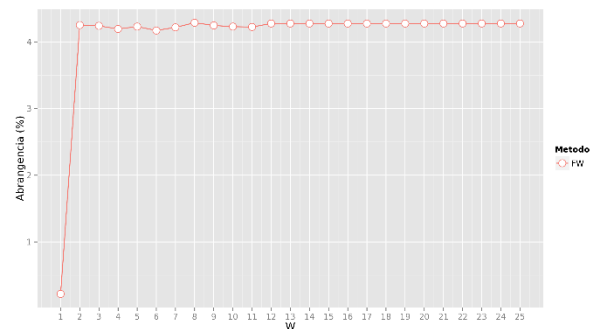


Figura 29 - Medida F_1 por quantidade de pesos W

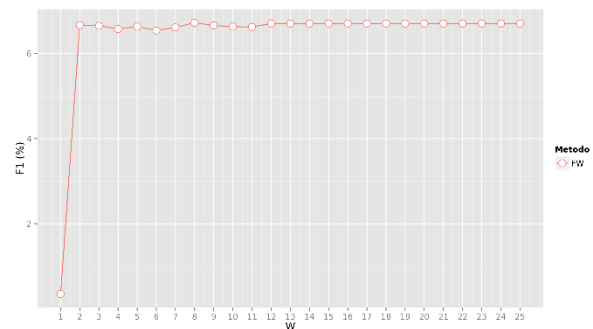
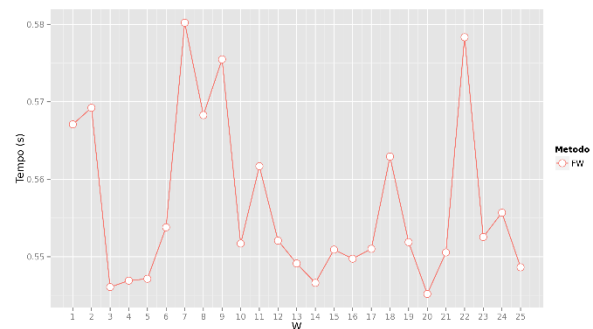


Figura 30 - Tempo de execução por quantidade de pesos W





6. Conclusão

Este Trabalho de Conclusão de Curso cumpriu seus objetivos e antecedeu aos requisitos estabelecidos no início do projeto. Foi elaborada uma biblioteca para sistemas de recomendação de produtos de comércio on-line e foi estabelecida uma respectiva análise de desempenho dos algoritmos de recomendação.

As avaliações de desempenho dos métodos propostos na biblioteca deste trabalho verificaram resultados já conhecidos no meio acadêmico. Em particular, a dependência entre qualidade de recomendação e tamanho da lista de sugestões se verificou (impacto de N).

Além disso, mostramos que um banco de dados com maior quantidade de avaliações (impacto de H) tem mais relevância que um banco de dados com mais usuários (impacto de T).

Outro resultado do trabalho foi a comprovação do fenômeno de *hidden feedback* (impacto de M). Mesmo que construamos métodos embasados na “avaliação positiva” dos usuários, esse parâmetro pode não ter tanta influência, visto que a maioria das avaliações dos clientes já são de fato positivas.

Também foi verificada a influência da quantidade de vizinhos mais próximos em algoritmos que usam essa metodologia colaborativa (impacto de k). Apesar de influenciar na qualidade da recomendação, esse parâmetro desempenha papel secundário.

Uma outra conclusão importante deste trabalho foi da importância de se escolher a priori o conjunto de atributos dos itens (impacto de F). A categorização excessiva dos itens pode ser maléfica para a recomendação, caso as *features* não tenham relevância para os usuários.

Avaliamos também diferentes medidas de distância entre os atributos (impacto de d_{ij}^f). A medida da diferença em valor absoluto foi comparada com outros índices, como o índice Jaccard, para uma lista de gêneros, e verificou-se que a distância L_1 resulta em melhor qualidade de recomendação. Vale ressaltar a importância da escolha das medidas de distância, visto seu impacto no desempenho do sistema.

Por fim, avaliamos também a quantidade de pesos dos atributos no método FW (impacto de W). Vimos que a quantidade de $w_f > 0$ não tem grande impacto na recomendação, visto que o valor dos pesos, em si, já é suficiente para alterar a qualidade das sugestões.

Apenas o método UP atingiu os requisitos funcionais em termos de precisão e abrangência, para uma combinação específica de parâmetros, como valores pequenos de N . Tanto para esse algoritmo quanto para o método FW, o desempenho é sensivelmente inferior ao relatado nos artigos de referência. O motivo por trás disso é a dissimilaridade entre os bancos de dados. Assim como foi

confirmado, o emprego de bases com mais recomendações r_{ui} influencia grandemente na qualidade das recomendações. Para se obter um benchmarking mais fiel, seria necessário utilizar o banco de dados dos autores de referência.

7. Trabalhos Futuros

A extensão desse Trabalho de Conclusão de Curso pode se dar de diversas maneiras, tanto na área acadêmica quanto na área empresarial. Seguindo o atual encaminhamento do projeto, a principal oportunidade do nosso trabalho é a criação de um serviço de um “Sistema de Recomendação nas Nuvens”.

Desejamos eliminar as restrições quanto a entrada e saída de dados, de forma que elas fossem completamente arbitrárias. O objetivo é que o usuário possa informar ao sistema como é formado sua base, e que todo o tratamento preliminar seja feito automaticamente.

É possível explorar também a construção de um driver que possibilite a conexão entre o sistema de recomendação e um banco de dados SQL, sem que seja necessária a etapa intermediária de arquivos csv para aquisição de dados. Em seguida, é importante elaborar um website para o sistema de recomendação e exportar toda a lógica para um servidor dedicado.

Outra melhoria desejada é a reconstrução dos métodos na linguagem de programação C, a fim de melhorar a performance computacional. Dessa forma, o serviço de “sistema de recomendação nas nuvens” estaria completo e poderia ser utilizado por comércio on-line reais.

Também seria desejável, para uma avaliação mais completa do trabalho, o emprego dos métodos computacionais em um banco de dados de um e-commerce real. Apesar de termos contatado diversas lojas de comércio online, devido a impedimentos administrativos, não obtivemos sucesso em firmar uma parceria com essas lojas.

No campo acadêmico, há muito espaço para melhorias nos algoritmos de recomendação. As metodologias de solução de cada um dos sistemas deveriam ser debatidas ao máximo, de modo a explorar casos de uso particulares e a propor mudanças e otimizações.

Faz-se necessário responder a perguntas como “O que acontece com itens ou usuários sem nenhuma avaliação?” e “Qual o desempenho dos métodos para outros bancos de dados?”.

AGRADECIMENTOS

Agradecemos ao professor Fábio Gagliardi Cozman pela orientação e inspiração.

Referências Bibliográficas

[1] EMARKETER. B2C Ecommerce Climbs Worldwide,



as Emerging Markets Drive Sales Higher. 2013. Disponível em: <<http://www.emarketer.com/Article/B2C-Ecommerce-Climbs-Worldwide-Emerging-Markets-Drive-Sales-Higher/1010004>>.

[2] EMARKETER. Global B2C Ecommerce Sales to Hit \$1.5 Trillion This Year Driven by Growth in Emerging Markets. 2014. Disponível em: <<http://www.emarketer.com/Article/Global-B2C-Ecommerce-Sales-Hit-15-Trillion-This-Year-Driven-by-Growth-Emerging-Markets/1010575>>.

[3] COOPERS, P. W. The Go-to-Market Revolution - Igniting Growth with Marketing, Sales, and Pricing. 2014. Disponível em: <https://www.bcgperspectives.com/content/articles/go_to_market_strategy_growth_go_to_market_revolution_igniting_growth_marketing_sales_pricing>.

[4] RICCI, L. R. F.; SHAPIRA, B. Introduction to recommender systems handbook. In: Recommender Systems Handbook. [S.l.]: Springer, 2011. p. 1–35.

[5] AMATRIAIN, X. Netflix Recommendations: Beyond the 5 stars. 2012. Disponível em: <<http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html>>.

[6] MARSHALL, M. Aggregate Knowledge raises \$5M from Kleiner, on a roll. 2006. Disponível em:

<<http://venturebeat.com/2006/12/10/aggregate-knowledge-raises-5m-from-kleiner-on-a-roll/>>.

[7] DAS, A. S. et al. Google news personalization: scalable online collaborative filtering. In: ACM. Proceedings of the 16th international conference on World Wide Web. [S.l.], 2007. p. 271–280.

[8] SARWAR, B. et al. Item-based collaborative filtering recommendation algorithms. In: ACM. Proceedings of the 10th international conference on World Wide Web. [S.l.], 2001. p. 285–29.

[9] SYMEONIDIS, P.; NANOPOULOS, A.; MANOLOPOULOS, Y. Feature-weighted user model for recommender systems. In: User Modeling 2007. [S.l.]: Springer, 2007. p. 97–106

[10] DEBNATH, S.; GANGULY, N.; MITRA, P. Feature weighting in content based recommendation system using social network analysis. In: ACM. Proceedings of the 17th international conference on World Wide Web. [S.l.], 2008. p. 1041–1042.

[11] LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, IEEE, v. 7, n. 1, p. 76–80, 2003.

[12] LOPS, P.; GEMMIS, M. de; SEMERARO, G. Advances in collaborative filtering. In: Recommender Systems Handbook. [S.l.]: Springer, 2011. p. 145–184.