UNIVERSIDADE FEDERAL DO AMAZONAS

INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LUDIMILA CARVALHO GONÇALVES

APRENDENDO FUNÇÕES DE PREVISÃO DE NOTAS EM MÉTODOS DE FILTRAGEM COLABORATIVA BASEADA EM USUÁRIO

Manaus

Março de 2013

UNIVERSIDADE FEDERAL DO AMAZONAS

INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LUDIMILA CARVALHO GONÇALVES

APRENDENDO FUNÇÕES DE PREVISÃO DE NOTAS EM MÉTODOS DE FILTRAGEM COLABORATIVA BASEADA EM USUÁRIO

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de Mestre em Informática.

Orientador: Prof. Marco Antônio Pinheiro de Cristo

Manaus

Março de 2013

Aos meus pais, Dione e Renato, que partiram cedo desta vida, mas deixaram em mim essa sede de aprendizado e de viver coisas novas que nunca se acaba.

...

Agradecimentos

Em primeiro lugar, minha eterna gratidão a Deus, por todas as bênçãos que recebi e por sempre me dar forças quando acredito que já não as tenho.

Agradeço à minha família por todo o carinho, que mesmo a distância, nunca deixou de dar apoio e estímulo ao longo de toda a minha tragetória acadêmica e também ao meu paciente e amado namorado, Cristian Rossi, que não só esteve comigo durante toda essa tragetória, como também cuidou de mim nos momentos mais difíceis (acadêmicos e da vida).

Não posso deixar de agradecer infinitamente à minha segunda família, meus sogros Elisabeth e Vilmar Rossi, e as minhas cunhadas Amanda e Maiane Rossi, que me receberam em sua casa com todo o amor como se fosse um membro real da família e me apoiaram incondicionalmente desde o início dessa jornada chamada Mestrado.

Essa jornada não poderia ser concluída sem o apoio daqueles que em todos os momentos estão presentes ao longo da vida: os amigos. Aos amigos que me acompanham desde a graduação: Adriana, André, Daniel, Diego, Helmer, Kleverson e Raíza. Aos adquiridos ao longo da caminhada acadêmica: Alexandre, David, Diana, Fortaleza, Juliana e Larissa. E, por fim, as amigas que não tem nada a ver com a vida acadêmica, porém tiveram igual ou até maior importância: Dani e Grazy, meu muito obrigada pelas horas de conversa, apoio, companhia no café, caronas e almoços comemorativos. Vocês são os melhores.

Ao meu orientador, Marco Cristo, obrigada pela compreensão, amizade e por todos os ensinamentos (que não foram poucos).

À FAPEAM e a Nokia/INdT pelo apoio financeiro concedido que fez com que pudesse me dedicar integralmente a este trabalho.

Por fim, não menos importante, ao IComp, professores e colaboradores, pelos ensinamentos, sem os quais nada disso seria possível, e por toda a dedicação na resolução rápida e eficaz de problemas burocráticos ao longo dos últimos anos.

"Sempre adiante; sempre mais longe; sempre mais alto."
(Léon Denis)

Resumo

A grande oferta de conteúdos na sociedade contemporânea torna difícil a tarefa de busca por informações que interessem aos usuários. Uma forma de lidar com tal sobrecarga de informações é prover ferramentas que recomendem para os usuários, dentre as informações alternativas, aquelas que devem ser de seu interesse. Tais ferramentas são os Sistemas de Recomendação (SR). As principais aplicações em SR se baseiam em duas técnicas, filtragem baseada em conteúdo e filtragem colaborativa. Dentre as duas, a filtragem colaborativa é a mais utilizada uma vez que, em geral, a estratégia que emprega, determinar grupos de usuários com interesses similares, é mais efetiva para capturar preferências. O problema de recomendação, como abordado em filtragem colaborativa, pode ser visto como um problema de previsão da preferência do usuário, normalmente representada por uma nota. Sistemas tradicionais prevêem esta nota através de uma equação de regressão obtida heuristicamente, envolvendo diversas evidências como nível de rigor do usuário e sua reputação. Como em qualquer estratégia heurística, não há nenhuma garantia que as equações usadas para a previsão sejam mais adequadas para um conjunto particular de dados, no sentido de minimizar o erro de previsão. Assim, neste trabalho, buscamos determinar se, em lugar de usar fórmulas heurísticas, não seria mais eficaz determinar automaticamente, por meio de uma técnica de aprendizagem de máquina, a melhor combinação das evidências disponíveis de forma a reduzir o erro de previsão. Nossos experimentos indicam que usando apenas evidências empregadas em métodos tradicionais, um método de regressão, como o proposto, pode alcançar resultados significativamente melhores que métodos tradicionais. Além disso, evidências como as notas que vizinhos atribuem ao item (como um todo ou individualmente) e as notas médias do usuário, do item e dos vizinhos possuíram melhor desempenho. Por fim, obtivemos ganhos de até 7% sobre o *baseline* com característica de confiança e de 6% sobre *baseline* sem uso de confiança.

Palavras-chave: Sistemas de Recomendação, Regressão, Aprendizado de Máquina, Previsão de Notas.

Abstract

The large offer of contents nowadays makes it hard to find relevant information. Recommender systems (RS) have been developed to tackle with such information overloading. Such systems are tools that recommend, from a large number of alternatives, the ones that the users will probably be interested in. The main RS applications are based on two approaches, content based filtering and collaborative filtering. Among them, collaborative filtering is the most used one since, in general, it employs a more effective strategy to capture user preferences: to determine groups of users with similar likes and dislikes. The recommendation problem, as viewed by collaborative filtering, can be viewed as the problem of predicting the preference of the user, normally represented as a rating. Traditional systems predict such ratings by means of manually-crafted regression equations obtained by combining different evidences such as: users reputation and its strictness level. As with any other heuristic strategy, there is no guarantee that the used equations are the best for a particular dataset in the sense of minimizing the prediction error. Thus, in this work, we intend to determine if it would be better to learn regression equations instead of using heuristically built ones. Such learned equations should be obtained by using a machine learning regression task to find the most effective combination of evidence on minimizing error. According to our experiments, a simple regression method is able to significantly outperform the best traditional equations using only evidence explored by those equations. Further, features like ratings that neighbors give to item (as all or individually) and user, item and neighbors average ratings have the best performance. Finally, we obtained gain of until 7% over the baseline with trust feature and gain of 6% over baseline without it.

Keywords: Recommender Systems, Regression, Machine Learning, Ratings prediction.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A grande oferta de conteúdos na sociedade contemporânea torna difícil a tarefa de busca por informações que interessem aos usuários. Em meio a uma sobrecarga de informações, surge a necessidade de ferramentas que possam processá-las de forma a descobrir quais são de interesse de um dado usuário. Dentre essas ferramentas, destacam-se os Sistemas de Recomendação (SR), que são ferramentas e técnicas que provêem sugestões personalizadas de itens para uso de um usuário [Ricci et al., 2011]. Item é o termo utilizado para o que o sistema recomenda para o usuário. Um Sistema de Recomendação normalmente é baseado em um tipo específico de item, como filmes ou livros e, em consequência, projetado de acordo com o mesmo.

Atualmente, SRs têm papel significativo em uma variedade de sítios muito populares na Internet, tais como Amazon[1], YouTube[2], Netflix[3], Yahoo![4], Tripadvisor[5], Apple[6], Last.fm[7] e IMDb[8] [Ricci et al., 2011]. Um exemplo da importância de tais sistemas é o fato da Netflix, uma companhia líder no segmento de aluguel de filmes nos EUA, ter promovido um concurso que concedeu um prêmio de um milhão de dólares à primeira equipe que fosse capaz de desenvolver um algoritmo de recomendação substancialmente melhor que o usado pela empresa [Bell & Koren, 2007].

A motivação para o interesse crescente em Sistemas de Recomendação são os inúmeros benefícios observados na prática [Ricci et al., 2011], dos quais citamos (a) o aumento nas vendas de itens, especialmente aqueles menos populares, o que contribui

[1] `http://www.amazon.com`
[2] `http://www.youtube.com`
[3] `http://www.netflix.com`
[4] `http://www.yahoo.com`
[5] `http://www.tripadvisor.com`
[6] `http://www.apple.com`
[7] `http://www.lastfm.com`
[8] `http://www.imdb.com`

para uma venda de maior diversidade; (b) o aumento na satisfação do usuário, com minimização de ofertas inadequadas; (c) o maior grau de compreensão de interesses e necessidades do usuário o que contribui, entre outras coisas, para (d) um alto nível de personalização e (e) fidelização dos clientes.

As principais aplicações em SR se baseiam em duas técnicas: filtragem baseada em conteúdo e filtragem colaborativa (FC). Na filtragem baseada em conteúdo, o usuário descreve diretamente os itens de seus interesses por meio de características que são usadas também para descrevê-lo (o que é conhecido como perfil do usuário). Já na abordagem de filtragem colaborativa, a recomendação é feita para um determinado usuário com base nos padrões de preferência de outros usuários. Dentre as duas, a filtragem colaborativa é mais utilizada uma vez que, em geral, melhores resultados são obtidos por métodos que buscam determinar grupos de usuários com interesses similares que os que buscam descrever preferências diretamente [Ricci et al., 2011]. Isso ocorre porque em muitos domínios (filmes, por exemplo), os usuários apresentam preferências muito diferentes em relação a itens muito similares ao passo que, em geral, concordam com as preferências de usuários com gostos similares aos seus.

A filtragem colaborativa é tradicionalmente classificada em duas abordagens: orientada aos usuários e baseada em itens. Na abordagem baseada em usuários, o sistema recomenda itens para um determinado usuário baseado no que outros usuários, similares a ele, gostaram. Na abordagem baseada em itens, o sistema avalia um item para um determinado usuário baseado na avaliação que este usuário forneceu para itens similares.

O problema de recomendação, como abordado em filtragem colaborativa, pode ser visto tanto como um problema de classificação (usuário vai gostar ou não deste item?) quanto previsão (o quanto o usuário vai gostar deste item?). Na prática, contudo, é tratado como um problema de previsão onde a preferência do usuário é representada por uma nota. Assim, dada uma base com avaliações anteriores dos usuários sobre um conjunto de itens, sistemas tradicionais preveêm a nota que o usuário atribuiria a um item novo, através de uma equação de regressão. Esta equação corresponde a uma fórmula fechada obtida heurísticamente e estruturada com base em características como, por exemplo, a nota média dada pelo usuário e a sua reputação. Ou seja, para obter a nota final do novo item, várias evidências são agregadas através de uma fórmula fechada, criada heurísticamente de forma prévia. Na literatura, muitas fórmulas têm sido propostas, explorando diferentes evidências, de diferentes formas [Ziegler & Golbeck, 2007; O'Donovan & Smyth, 2006, 2005].

1.1 Problema e questões de pesquisa

Como em qualquer estratégia heurística criada por um ser humano, não há nenhuma garantia que fórmulas fechadas usadas para a previsão de preferências em filtragem colaborativa sejam mais adequadas para um conjunto particular de dados, no sentido de minimizar o erro de previsão. Logo, considerando que é possível determinar a nota exata de um certo item, dado o conjunto completo de evidências, algumas questões surgem naturalmente:

- Em lugar de usar fórmulas heurísticas, não seria mais eficaz determinar automaticamente, por meio de uma técnica de aprendizagem de máquina, a melhor combinação das evidências disponíveis de forma a reduzir o erro de previsão?

- Evidências derivadas das originais não poderiam ser mais eficazes em termos de facilitar a previsão das notas?

- Dentre as evidências usadas para realizar a previsão (sejam as tradicionalmente usadas, sejam as novas derivadas para este problema), quais são as mais importantes, considerando a sua contribuição para a minimização do erro?

- Qual o conjunto mínimo de evidências, entre as estudadas, mais eficaz para o problema dado?

1.2 Objetivos

1.2.1 Objetivo Geral

Avaliar o uso de métodos de regressão, em oposição ao uso de fórmulas fechadas tradicionalmente usadas e obtidas de forma heurística, para aprender automaticamente como prever notas que usuários dariam para itens em Sistemas de Recomendação. A avaliação será feita com base em evidências usadas em métodos tradicionais, bem como novas evidências a serem propostas neste trabalho.

1.2.2 Objetivos Específicos

1. Selecionar e implementar as evidências usadas em fórmulas fechadas, na literatura, com a proposição de novas evidências;

2. Implementar ou obter implementações dos métodos padrões de filtragem colaborativa a serem usados como base para comparação, usando diferentes fórmulas fechadas, de forma a explorar todas as evidências selecionadas;

3. Implementar métodos de regressão que explorem as várias evidências selecionadas;

4. Comparar as várias estratégias, considerando diferentes conjuntos de evidências, de forma a determinar que abordagem é melhor e quais as evidências mais importantes.

1.3 Metodologia

A realização desta pesquisa se deu com a execução das seguintes tarefas:

1. Levantamento bibliográfico sobre o estado da arte relacionado ao problema proposto, com definição do foco da pesquisa.

2. Obtenção de coleções de referências para experimentação.

3. Seleção e implementação de métodos tradicionais de filtragem colaborativa.

4. Instrumentação dos métodos básicos para a extração de evidências usadas tradicionalmente e implementação das novas evidências a serem avaliadas.

5. Aplicação de método de regressão às evidências estudadas.

6. Análise de resultados e conclusão.

Como resultado do estudo bibliográfico, escolhemos como objeto de nosso estudo métodos de filtragem colaborativa, por serem muito difundidos. Entre estes métodos, a abordagem baseada em itens é amplamente utilizada devido à sua simplicidade, eficiência e capacidade de produzir recomendações precisas e personalizadas. Porém, métodos orientados a usuários são mais recomendados em situações em que (a) o sistema tem mais itens que usuários e, portanto, informação relativa à similaridade entre usuários é mais confiável [Good et al., 1999], (b) os itens mudam constantemente, como no caso de recomendação online, o que torna o cálculo de pesos mais caro [Ricci et al., 2011] e (c) se deseja fornecer recomendações mais surpreendentes.

Considerando a grande variedade de métodos e fórmulas na literatura, não seria possível oferecer uma comparação completa. Assim, restringimos nosso estudo a fórmulas fechadas usadas no contexto de métodos de filtragem colaborativa baseada em

usuários. Em particular, utilizamos a implementação mais aplicada na indústria para este método, obtida através de uma ferramenta publicamente disponível e largamente usada, o Apache Mahout [Apache Software Foundation, 2011]. Adicionalmente, expandimos essa implementação para que suportasse a reputação dos usuários, uma classe de evidências que foi demonstrada útil no contexto de filtragem colaborativa [O'Donovan & Smyth, 2005]. Ambas as implementações foram instrumentadas para que fosse possível a extração das várias evidências usadas nos métodos estudados.

Quanto ao tratamento do problema de recomendação como um problema de previsão, nossa idéia é representar cada nota a ser dada como um evento envolvendo um item i (a ser avaliado), um usuário u (para quem a recomendação deve ser feita) e o conjunto $\mathcal{V}_u$ de usuários mais similares a u (a sua vizinhança) de acordo com uma métrica de similaridade. O problema de recomendação consiste, então, em determinar que nota o usuário u daria ao item i considerando informações do passado de u, i e os vizinhos em $\mathcal{V}_u$. Este problema pode ser pensado como o de encontrar uma função de regressão. Dado que o custo de aprendizado desta função é proporcional ao número de instâncias e variáveis consideradas, nós optamos por métodos de regressão mais simples, Regressão Linear e M5P, já que não encontramos trabalhos na literatura que utilizem métodos de aprendizagem de máquina para resolução do problema em questão.

Para avaliar os métodos implementados, usamos uma coleção de referência publicamente disponível, o MovieLens[9]. Esta consiste em uma coleção de filmes com 10.000 avaliações sobre 1682 filmes feitas por 943 usuários, onde cada usuário avaliou pelo menos 20 filmes. Informação de reputação entre os usuários nessa coleção foi derivada de acordo com o método proposto em [O'Donovan & Smyth, 2005].

1.4 Resultados

Nossos resultados mostram que métodos de aprendizagem possuem melhores resultados que fórmulas fechadas. Mais especificamente, nosso principal resultado, mostra que o método de regressão M5P obtém ganhos significativos sobre o método heurístico *baseline* Resnick et al. [1994] quando utiliza a mesma abordagem e possui ganhos maiores quando utiliza informação de reputação. Nesse caso, também obtivemos ganhos sobre o método heurístico *baseline* O'Donovan & Smyth [2006]. Também observamos que os melhores conjuntos de características a serem utilizados na aprendizagem estão relacionados com notas médias e passadas. Mais detalhes sobre os resultados obtidos

[9] `http://www.movielens.org`

são apresentados no Capítulo 5.

1.5 Estrutura da Dissertação

Além deste capítulo introdutório, este trabalho está organizado como segue. No Capítulo 2, descrevemos os conceitos básicos necessários para compreensão deste trabalho, bem como uma revisão da literatura, o que inclui métodos para calcular reputação de usuários, bem como para recomendação baseada em aprendizado de máquina e aprendizado de *ranking*. No Capítulo 3, apresentamos a representação usada para a nota a ser prevista, baseada em características dos usuários, itens e vizinhança. No Capítulo 4, apresentamos os experimentos realizados e os resultados obtidos. Finalmente, no Capítulo 5, apresentamos nossas conclusões e sugestões de próximos passos na pesquisa.

Capítulo 2

Fundamentos

Neste capítulo, apresentamos conceitos básicos para a compreensão dos métodos estudados, bem como uma revisão da literatura relacionada ao nosso problema. Os conceitos discutidos incluem Sistemas de Recomendação (filtragem colaborativa e reputação de usuários), métodos de aprendizado de máquina, técnicas de regressão (linear e M5P) e metodologia de avaliação de SRs.

2.1 Sistemas de Recomendação

Um Sistema de Recomendação (SR) é um sistema que combina várias técnicas computacionais para selecionar itens personalizados com base nos interesses dos usuários. Tais sistemas surgiram como resposta à dificuldade das pessoas em escolher entre uma grande variedade de produtos e serviços alternativos que lhe são apresentadas. Entre os itens normalmente recomendados podemos citar livros, filmes, notícias, música, vídeos, anúncios, publicidade, páginas da internet, produtos, etc. A área de Sistemas de Recomendação emergiu como uma área de pesquisa independente no meio da década de 90, quando pesquisadores passaram a focar em problemas de recomendação que explicitamente invocavam estruturas de avaliação baseadas em notas (*ratings*).

Formalmente, um sistema de recomendação pode ser definido como segue. Seja $\mathcal{U}$ o conjunto de todos os usuários de um determinado sistema, e seja $\mathcal{I}$ o conjunto de todos os possíveis itens que podem ser recomendados como livros, filmes, restaurantes etc. Seja μ a função utilidade que mede o quão útil é um determinado item i para um determinado usuário u, i.e., $\mu : \mathcal{U} \times \mathcal{I} \rightarrow R$, onde R é um conjunto totalmente ordenado. Então, para cada usuário $u \in \mathcal{U}$, procura-se um item $i_u \in \mathcal{I}$ que maximiza

a utilidade do usuário u. Isto pode ser expresso pela Equação 2.1.

$$\forall_{u \in \mathcal{U}} \; i_u = \arg\max_{i \in \mathcal{I}} \mu(u, i) \tag{2.1}$$

O problema de SRs é que a função μ não é definida para todo espaço $\mathcal{U} \times \mathcal{I}$, mas apenas em um subconjunto deste. Isto significa que μ precisa ser extrapolada para todo o espaço $\mathcal{U} \times \mathcal{I}$. Deste modo, o algoritmo de recomendação deve ser capaz de prever as avaliações não realizadas para os pares usuário-item e fazer recomendações baseadas nestas previsões.

A previsão de avaliações de itens ainda não avaliados pode ser feita de diferentes formas. SRs são classificados de acordo com o método, em geral, nas seguintes categorias:

- Métodos baseados em conteúdo: o usuário receberá recomendações de itens similares a itens preferidos no passado;
- Filtragem colaborativa: o usuário receberá recomendações de itens baseadas em gostos de pessoas com gostos similares aos dele;
- Métodos Híbridos: estes métodos combinam tanto estratégias de recomendação baseadas em conteúdo quanto estratégias baseadas em colaboração.

Dentre esses métodos, os baseados em filtragem colaborativa são muito usados por serem fáceis de implementar e muito efetivos em uma variedade de cenários. Como descrito anteriormente, métodos de filtragem colaborativa podem ser orientados aos usuários ou baseados em itens[1]. No primeiro caso, o sistema recomenda itens para um determinado usuário baseado no que outros usuários, similares a ele, gostaram. Nesse caso, usuários são similares na medida em que gostam dos mesmos itens. No segundo caso, o sistema avalia um item para um determinado usuário baseado na avaliação que este usuário forneceu para itens similares. Nesse caso, dois itens são considerados similares na medida em que mais usuários (em comum) os avaliam da mesma forma.

Neste trabalho, estamos particularmente interessados em sistemas colaborativos baseados em usuário. Embora, em geral, eles sejam preteridos em relação a métodos baseados em itens, há cenários em que o seu uso ainda é recomendado. Este é o caso de sistemas em que (a) a informação relativa à similaridade entre os usuários é mais confiável Good et al. [1999], (b) os itens mudam constantemente, como no caso de recomendação online, o que torna o cálculo de pesos mais caro Ricci et al. [2011] e (c)

[1] Abordagens baseadas em fatores latentes Ricci et al. [2011] podem usar um arcabouço baseado em usuários, itens ou ambos, não sendo discutidas nesta dissertação

se deseja correr o risco de fornecer recomendações mais surpreendentes. Isso porque no método baseado em itens, serão recomendados apenas itens similares ao que o usuário já viu, ou seja, recomendações seguras, porém previsíveis. Na baseada em usuários, é possível que um usuário u_1 seja similar a um usuário u_2 em um nicho de preferências, porém u_2 também gosta de itens fora daquele nicho. Logo, recomendações diferentes podem ser feitas para u_1.

Nas próximas seções, sistemas de filtragem colaborativa baseada em usuários e extensões que incorporam informação de reputação dos usuários são descritos em mais detalhes.

2.1.1 Filtragem Colaborativa baseada em Usuário

A filtragem colaborativa baseada em usuário (FCU) consiste em recomendar para um usuário u itens que u não conhece e que são muito populares entre os usuários mais similares a u. A similaridade entre os usuários é calculada com base nos itens que eles consumiram em comum.

Mais formalmente, o cerne do FCU consiste de uma regressão cujo objetivo é prever a nota que o usuário u daria ao item i ($\hat{r}_{ui}$). Para tanto, uma métrica de similaridade é usada para identificar os k usuários que são mais similares a u e que deram nota para i ($S^k(i,u)$). O valor previsto é a média ponderada, pela similaridade (s_{uv}), das notas dos usuários vizinhos (r_{uv}), ajustados pela nota média $\overline{r}$.

$$\hat{r}_{ui} = \overline{r}_u + \frac{\sum_{v \in S^k(i,u)} S_{uv}(r_{vi} - \overline{r}_v)}{\sum_{v \in S^k(i,u)} S_{uv}} \tag{2.2}$$

onde a similaridade S_{uv} é calculada com o coeficiente de correlação de Pearson (PCC - *Pearson Coeficient Correlation*) Herlocker et al. [2004] Ricci et al. [2011] como segue.

$$PCC(u,v) = \frac{\sum_{i \in \mathcal{I}_u v}(r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in \mathcal{I}_{uv}}(r_{ui} - \overline{r}_u)^2 \sum_{i \in \mathcal{I}_u v}(r_{vi} - \overline{r}_v)^2}} \tag{2.3}$$

onde $\mathcal{I}_{uv}$ é o conjunto de itens avaliados por u e também por v. Na prática, apenas usuários com correlação S_{uv} positiva e que deram nota pra i são considerados. A Equação 2.2 se deve a Resnick et al. [1994]. Note que a Equação 2.2 envolve os seguintes atributos: notas (normalizadas pela média, o que implica em nota média do usuário e do vizinho, além da nota absoluta dada por um vizinho a um item) e similaridade entre usuário e vizinhos. Variantes muito usadas desta equação ainda consideram as popularidades dos usuários, vizinhos e itens como fatores de correção. Ao longo deste trabalho, nos referimos à implementação desta equação como FCU-Resnick.

Com intuito de realizar o estudo em uma implementação escalável de um SR, aplicável em ambiente industrial, a implementação que usamos como FCU-Resnick é a disponibilizada pelo projeto Apache Mahout Apache Software Foundation [2011]. Este projeto fornece implementações escaláveis de vários algoritmos de aprendizagem de máquina e mineração de dados, entre as quais, filtragem colaborativa. As implementações foram feitas no topo de uma infra-estrutura distribuída baseada em *map-reduce* Dean & Ghemawat [2008] e são otimizadas para atingir bom desempenho usando uma ou múltiplas máquinas.

O tamanho k da vizinhança, apresentada na Equação 2.2, é determinado pelo método kNN e é apresentado na seção que segue.

2.1.1.1 *k*-Vizinhos mais Próximos

Conhecido como classificador baseado em instâncias, o método dos k vizinhos mais próximos (kNN - *k-Nearest Neighbor*) classifica um novo item a partir de um conjunto de treinamento armazenado Cover & Hart [1967]. Assim, dada uma nova instância, esta é comparada com exemplos de treino, usando uma métrica de distância, de forma que a classe da nova instância é determinada pelas classes das k mais similares.

Métodos de vizinhança são a base de estratégias de filtragem colaborativa, as mais usadas em SRs. Bell & Koren [2007] mostram que o sucesso do kNN se deve a três principais componentes que caracterizam o método: (1) normalização de dados, (2) seleção de vizinhos e (3) determinação dos pesos de interpolação. Segundo os autores, enquanto há poucas diferenças entre os métodos de seleção de vizinhos, os outros dois aspectos são importantes para o melhor funcionamento do método.

2.1.2 Reputação de Usuários

Nem sempre os usuários mais parecidos com um certo usuário u, para o qual se pretende fazer recomendaçãoes, são confiáveis. Muitas vezes, usuários que fazem revisões em SRs estão interessados em divulgar um certo item e, portanto, sua opinião sobre ele não é isenta. Outras vezes, estes usuários podem simplesmente ter um gosto distinto do de u em certo segmento de itens. Independente da razão, é interessante que SRs considerem a reputação dos usuários vizinhos, sob a perspectiva de u.

SRs considerando relações de confiança entre usuários (reputação) foram propostos para prover recomendações mais personalizadas e precisas aos usuários. Eles partem da intuição de que usuários preferem recomendações de outros usuários, nos quais eles confiam. Isto é obtido através de uma rede de confiança que expressa o quanto os membros de uma comunidade confiam uns nos outros Ricci et al. [2011].

Muitos trabalhos tem sido propostos tanto para obter informação de reputação quanto para aplicá-la em SRs.

Por exemplo, Ma et al. [2009] propõem um arcabouço de otimização baseado na análise de fatores com fatoração de matriz para incorporar relações de confiança e desconfiança em SRs. A complexidade do método é linear nas observações das avaliações e experimentos mostram que ele supera outros algoritmos do estado da arte. A partir das análises experimentais, os autores concluem que informação de desconfiança é mais importante que a de confiança. O trabalho abordou as informações de confiança e desconfiança separadamente.

Nas seções a seguir, descrevemos variações da equação de regressão da FCU (Equação 2.2) que levam em conta informação de reputação.

2.1.2.1 Filtragem Colaborativa baseada em Confiança (*trust*) (FCC)

Dada a informação de confiança entre os usuários u e v, c_{uv}, e o conjunto de usuários na rede de confiança de u, $(S^C(u))$, de acordo com Ziegler & Golbeck [2007] a Equação 2.2 pode ser re-escrita como:

$$\hat{r}_{ui} = \overline{r}_u + \frac{\sum_{v \in S^C(u)} t_{uv}(r_{vi} - \overline{r}_v)}{\sum_{v \in S^C(u)} c_{uv}} \tag{2.4}$$

Esta estratégia é possível uma vez que há uma correlação entre similaridade e confiança. Note que os pesos c_{uv} são obtidos das relações entre os usuários em uma rede de confiança obtidas por meio de operações de propagação e agregação.

Propagação e agregação de confiança são as principais partes da construção de métricas de confiança, que visam estimar a confiança entre dois usuários desconhecidos da rede. São mecanismos para estimar a confiança transitivamente pela computação de quanto um usuário u confia em um usuário v_1, dado que u confia em v_2 e v_1 possui um valor de confiança para v_2 (propagação) e, pela combinação dessas várias estimativas de confiança em um valor final (agregação) Ricci et al. [2011]. Note que esta equação substitui a similaridade entre usuário e vizinho pela confiança entre usuário e vizinho.

2.1.2.2 Filtragem baseada em Confiança (*trust*) (FbC)

O método anterior requer que os usuários forneçam explicitamente informação de confiança. Como esta informação nem sempre está disponível, métodos que inferem estimativas de confiança a partir das informações disponíveis podem ser uma solução melhor O'Donovan & Smyth [2006].

Em particular, dado o método proposto por O'Donovan & Smyth [2005], é possível calcular duas métricas de confiança, uma refletindo a confiabilidade geral de um usuário v (confiança em nível de perfil) e outra refletindo a confiança de v relacionada com um item i (confiança em nível de item). Estes métodos se baseiam na intuição de que um usuário que forneceu boas recomendações no passado (do ponto de vista do usuário u, que vai receber as recomendações) é mais confiável que um que não teve tão bom desempenho.

Note então que é possível criar um conjunto $S^{kC}(u)$ que engloba todos os usuários que tem correlação positiva com u e cuja confiança em nível de perfil (ou item) é maior que um certo limiar c_{min}. Dado este conjunto $S^{kC}(u)$, a Equação 2.2 pode ser re-escrita como:

$$\hat{r}_{ui} = \overline{r}_u + \frac{\sum_{v \in S^{kC}(u)} S_{uv}(r_{vi} - \overline{r}_v)}{\sum_{v \in S^{kC}(u)} S_{uv}} \tag{2.5}$$

Nesta equação, os valores de confiança são usados como filtro, de forma que apenas vizinhos confiáveis participam do processo de recomendação. Das variantes de métodos propostos em O'Donovan & Smyth [2006], a que obteve melhor desempenho em nossos experimentos foi a dada pela Equação 2.5 utilizando a abordagem de confiança em nível de perfil. Ao longo deste trabalho, nos referimos a esta implementação como FbC-Donovan.

Note que esta equação envolve os mesmos atributos de FCU. Contudo usa, implicitamente, informação de confiança sem abrir mão de informação de similaridade.

2.2 Métodos de Previsão Numérica baseados em Regressão

Neste trabalho, o problema de previsão de nota é modelado como uma regressão numérica. Considerando que uma grande variedade de evidências serão avaliadas, incluindo algumas relacionadas aos vizinhos mais similares ao usuário alvo, nosso estudo foi focado em técnicas de regressão que não demandassem tantos recursos computacionais, ou seja, as lineares ou combinações de lineares[2]. Assim, escolhemos os métodos de regressão linear e M5P (*model tree*) fornecidos com o pacote Weka Hall et al. [2009]. Esses métodos são descritos nas seções a seguir.

[2]De fato, ao todo, avaliamos outros três métodos de regressão não linear: a regressão baseada em vetores de suporte (SVR) Drucker et al. [1997], o IBk Aha et al. [1991] com valores de $k = \{1, 5, 10, 20, 30\}$ e redes neurais de múltiplas camadas Rosenblatt [1962] Widrow & Lehr [1990]. Em todos os casos, ganhos não justificaram os altos custos de treinamento observados.

2.2.1 Regressão Linear

A idéia da regressão linear é que a variável alvo (em nosso caso, a nota r) é dependente de um conjunto de variáveis observadas $X_1, ..., X_m$ (em nosso caso, evidências como nota média do usuário, similaridade pro usuário alvo, etc). Assim, temos:

$$r = \beta_0 + \sum_{j=1}^{m} \beta_j X_j + \epsilon \tag{2.6}$$

onde ϵ é o erro (variável aleatória não observada com média 0 e variância σ^2). A Equação 2.6 é chamada modelo de regressão linear com multiplas variáveis. Os parâmetros β_i são desconhecidos e têm variância de erro ($\sigma^2 > 0$) desconhecida. O objetivo é determinar $\boldsymbol{\beta}$ e σ^2.

Para prever o valor de r na Equação 2.6 é necessário determinar uma função $f(\boldsymbol{X}), \boldsymbol{X} = \{X_1, ..., X_m\}$. Dada uma função *perda* que indica o quanto a função $f(\boldsymbol{X})$ é diferente de r, o risco esperado ao adotar f pra prever r é dado por:

$$risco(f) = E(perda(r, f(\boldsymbol{X})) \tag{2.7}$$

Usando como função de perda a média do quadrado do erro, temos:

$$risco(f) = E(r - f(\boldsymbol{X}))^2 \tag{2.8}$$

Há várias formas de resolver a Equação 2.8. De forma geral, a equação $f(\boldsymbol{X})$ pode ser escrita como:

$$f(\boldsymbol{X}) = \beta_0 + \sum_{j=1}^{m} \beta_j X_j = \beta_0 + \boldsymbol{X}^T \boldsymbol{\beta} \tag{2.9}$$

Que, minimizada para reduzir a função de perda (Equação 2.8), tem os seguintes valores estimados para β_0 e $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{\Sigma}_{\mathrm{XX}})^{-1} \boldsymbol{\Sigma}_{\mathrm{Xr}} \tag{2.10}$$

$$\hat{\beta}_0 = \mu_r - \boldsymbol{\mu}_X^T \hat{\boldsymbol{\beta}} \tag{2.11}$$

onde $\boldsymbol{\Sigma}_{\mathrm{XX}}$ é a matriz de covariância de $\boldsymbol{X}$, $\boldsymbol{\Sigma}_{\mathrm{Xr}}$ é a matriz de covariância de $\boldsymbol{X}$ e r, μ_X e μ_r são as médias de $\boldsymbol{X}$ e r. Note que nenhum desses elementos são realmente conhecidos para a população inteira. Assim, eles precisam ser estimados. Uma estratégia comum é estimá-los de uma amostra de tamanho n. Neste caso, $\boldsymbol{\mu}_X$ e μ_r são tomados como as

médias de $\boldsymbol{X}$ e r na amostra. Mais especificamente:

$$\boldsymbol{\mu}_X = \overline{\boldsymbol{X}} = \frac{1}{n}\sum_{j=1}^{n} X_j,\ \mu_r = \bar{r} = \frac{1}{n}\sum_{j=1}^{n} r_j \tag{2.12}$$

Para estimar as matrizes de covariância, é necessário definir as matrizes $\mathcal{X}_c$ e $\mathcal{R}_c$ que correspondem as formas centradas na média de $\boldsymbol{X}_1, ..., \boldsymbol{X}_n$ e r_i, ou seja, os valores originais subtraídos das suas médias. Assim, $\boldsymbol{\Sigma}_{\mathrm{XX}} = n^{-1}\mathcal{X}_c^T\mathcal{X}_c$ e $\boldsymbol{\Sigma}_{\mathrm{Xr}} = n^{-1}\mathcal{X}_c^T\mathcal{R}_c$. Logo:

$$\hat{\boldsymbol{\beta}}^* = \left(\mathcal{X}_c{}^T\mathcal{X}_c\right)^{-1}\mathcal{X}_c{}^T\mathcal{R}_c \tag{2.13}$$

$$\hat{\beta}_0{}^* = \bar{r} - \overline{\boldsymbol{X}}^T\hat{\boldsymbol{\beta}}^* \tag{2.14}$$

2.2.2 M5P (*Model Tree*)

M5P Wang & Witten [1997] é um algoritmo que combina uma árvore de decisão tradicional com a possibilidade de funções de regressão nos nós. Primeiro, um algoritmo tradicional de indução de árvore Hall et al. [2009] é usado para construir uma árvore. O *critério de divisão* dos nós é minimizar o erro de previsão de uma função de regressão envolvendo o subconjunto de instâncias relacionadas com o nó. Este processo é realizado até que um *critério de parada* seja atingido. Uma vez que uma árvore é obtida, ela é submetida a um *processo de poda* com intuito de ser mais geral. A árvore resultante é então modificada por um *processo de suavização* que tenta eliminar discontinuidades bruscas observadas entre as subárvores. M5P gera modelos compactos e relativamente compreensíveis.

Em nosso caso particular, para determinar qual atributo melhor divide uma porção T dos dados de treino é usado como *critério de divisão* o desvio padrão da classe de valores em T. O atributo que maximiza a redução do erro é escolhido para divisão no nó. A redução do erro SDR (*standard deviation reduction*) é calculada pela Equação 2.15:

$$SDR = sd(T) - \sum_j \frac{|T_j|}{|T|} \times sd(T_j) \tag{2.15}$$

onde $T_1, T_2, ...$ são os conjuntos que resultam da divisão do nó de acordo com a escolha do atributo e $sd(T)$ é o desvio padrão da classe de valores.

O *processo de poda* empregado consiste em parar o processo de divisão quando a classe de valores que atinge o nó varia pouco.

A poda utiliza uma estimativa, em cada nó, de erro esperado para os dados de teste. Primeiro, a diferença absoluta entre a nota prevista e o valor real da classe é medida sobre cada uma das instâncias de treino que chegam no nó. Devido ao fato de a árvore ser construída para esse conjunto de dados, esta média sobre o conjunto subestimará o erro esperado para casos não vistos. Para compensar, é multiplicado pelo fator $(n+v)/(n-v)$, onde n é o número de instâncias de treinamento que chegam no nó e v é o número de parâmetros no modelo linear que possuem a classe de valor neste nó.

O erro esperado para os dados de teste é calculado como descrito anteriormente, usando um modelo linear para previsão. Devido ao fator de compensação $(n+v)/(n-v)$, o modelo linear pode ser simplificado por termos retirados para minimizar o erro estimado. Apagar um termo diminui o fator de multiplicação, que pode ser suficiente para compensar o inevitável aumento da média de erro sobre as instâncias de treinamento.

Por fim, uma vez que cada nó interno possui um modelo linear, a árvore é podada de volta das folhas ao passo que o erro estimado diminui. O erro esperado para o modelo linear no nó, é comparado com o erro esperado da subárvore anterior. Para calcular esta última, o erro de cada ramo é combinado em um valor médio para o nó, somando o peso de cada ramo multiplicado pelo número de instâncias de treino que caíram e combinando as estimativas de erro linearmente utilizando esses pesos.

Ao final, é feito um *processo de suavização* para compensar problemas que ocorrem entre modelos adjacentes nas folhas da árvore podada, devido a modelos contruídos a partir de poucas instâncias de treino. O procedimento utiliza o modelo da folha para calcular a nota prevista. Em seguida, filtra este valor ao longo do caminho de volta para a raiz, suavizando-o em cada nó através da sua combinação com a previsão do modelo linear do nó corrente. Este cálculo se dá da seguinte forma:

$$r' = \frac{nr + kq}{n + k} \tag{2.16}$$

onde r' é a previsão passada para o próximo nível de nó, r é a previsão passada pelo nó anterior, q é o valor previsto pelo modelo no nó corrente, n é o número de instâncias de treinamento que atingem o nó anterior e k é uma constante[3]. Segundo Wang & Witten [1997] a suavização aumenta substancialmente a acurácia das previsões.

[3]Valor padrão equivale a 15

2.3 Avaliação

Nesta seção, descrevemos a metodologia de avaliação que empregamos nesta dissertação, bem como a métrica usada para avaliar a precisão dos métodos de previsão estudados, o RMSE.

2.3.1 Metodologia de Avaliação

Com intuito de obter resultados mais confiáveis, em todos os experimentos comparativos, nós usamos a validação cruzada de n partições Hall et al. [2009]. Neste método, cada base de dados é dividida em n partições, tal que, em cada rodada de experimentação, uma partição diferente é usada como conjunto de teste enquanto o restante das instâncias é usado como conjunto de treino. Nós usamos as mesmas divisões de treino e teste em todos os experimentos.

Adicionalmente, para todas as comparações mostradas nesta dissertação, nós usamos o teste de posto sinalizado de Wilcoxon Wilcoxon [1945] para determinar se as diferenças observadas em RMSE são estatisticamente significativas. Este é um teste pareado não paramétrico que não assume qualquer distribuição particular dos valores testados. Nós consideramos estatisticamente significativos apenas resultados com nível de confiança acima de 95%.

2.3.2 Raíz do Erro Quadrado Médio (RMSE - *Root Mean Squared Error*)

Para calcular o erro entre as previsões realizadas e os valores reais, nós utilizamos a métrica padrão raíz do erro quadrado médio (RMSE). Esta métrica é calculada como dado na Equação 2.17.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{r}_i - r_i)^2} \tag{2.17}$$

onde n é o número de instâncias de teste, $\hat{r}_i$ é a nota prevista pelo método avaliado para a instância de teste i e r_i é a nota real da instância i.

2.4 Trabalhos Relacionados

Considerando que nosso método tem por objetivo o aprendizado da função de previsão e esta pode ser vista como uma função de *ranking*, nesta seção apresentamos alguns

trabalhos em SRs baseados na estratégia de aprendizado de *ranking*.

2.4.1 Aprendizado de Ordenação

Aprendizado de ordenação (LTR - *Learning to rank*) é uma outra abordagem de aprendizado de máquina, muito usada em recuperação de informação, que vem sendo proposta também para problemas de recomendação, especialmente, os baseados em filtragem colaborativa. Nesta abordagem, um modelo ou função de *ranking* é aprendido a partir das informações de preferência (ordem relativa) entre as instâncias (*list-wise*). Como esta técnica consiste em retornar, como resposta, uma lista de itens em certa ordem, ela se adéqua bem ao problema de recomendação de listas de itens, embora não seja muito apropriada para a previsão da nota de um item (*point-wise*). Em seguida, descrevemos alguns trabalhos que usam LTR em SRs.

Em Shi et al. [2010], os autores propõem o método ListRank-MF para um SR de filtragem colaborativa baseada em item que combina um algoritmo de LTR com a técnica de fatoração matricial para representar usuários e itens através de características latentes. O ListRank-MF utiliza a entropia cruzada das melhores probabilidades dos itens nas listas de exemplos de treinamento e nas listas de classificações a partir da minimização do modelo de classificação de função de perda, que representa a incerteza entre a lista de treinamento e a de saída produzida pelo método, é gerada uma lista com *ranking* de itens que desfruta da vantagem de possuir baixa complexidade. O desempenho do método proposto é comparado com a abordagem tradicional de filtragem colaborativa baseada em item e com um método estado-da-arte, o CoFiRank. Na maior parte dos casos, o ListRank-MF supera significativamente os demais. Devido ao fato de ter tido comportamento linear no número de avaliações observadas, dada uma matriz usuário-item, o método pode ser aplicado para o uso em larga escala, envolvendo coleções reais.

Em Liu & Yang [2008], os autores propõem um método de *ranking* de itens para filtragem colaborativa que não passa pela etapa intermediária de previsão de nota, gerando diretamente o *ranking* de itens. Primeiro, é descrita uma medida de similaridade para comparar dois *rankings* de usuários (*Kendall Rank Correlation Coeficient*), envolvendo conjuntos de itens usados para determinar conjuntos de usuários que possuem as mesmas preferências em relação ao um usuário alvo. Em seguida, são propostos dois métodos para criação de *rankings* de itens baseados no conjunto de usuários similares. O primeiro, é um algoritmo guloso que maximiza o valor da função de preferência que procura nos *rankings* o melhor deles. O segundo, é um novo modelo de caminhada aleatória, definido através da informação de preferências do usuário, baseado em cadeia

de Markov. Os resultados reportados mostram que o arcabouço proposto supera os algoritmos de filtragem colaborativa existentes.

2.4.2 Métodos de Recomendação baseado em Aprendizado de Máquina

Nesta seção, descrevemos o uso de métodos de aprendizado de máquina em SRs. Os trabalhos aqui relacionados são provenientes da revisão apresentada em Ricci et al. [2011], que mostra trabalhos do estado-da-arte relacionados a SR como um todo. Note que nenhum dos métodos apresentados nesta seção tem o objetivo de aprender a função de previsão da nota sendo, portanto, diferentes do nosso.

2.4.2.1 Árvores de Decisão

Árvores de decisão Quinlan [1986] (AD) são classificadores baseados em árvores formadas por dois tipos de nós: (1) nós de decisão, onde um atributo-valor é testado para determinar qual subárvore se aplica a classificação; e (2), nós folhas, onde são indicadas as classes dos itens. Uma das principais vantagens da árvore de decisão é a rapidez na classificação de instâncias desconhecidas.

ADs foram empregadas de várias formas em SRs. Por exemplo, Bouza et al. [2008] modelaram as preferências do usuário (as folhas da AD) a partir de características de conteúdo (os nós de decisão). Embora essa abordagem seja interessante do ponto de vista teórico, na prática, a precisão do sistema é pior que a recomendação baseada na média das avaliações.

2.4.2.2 Classificadores Baseados em Regras

Classificadores baseados em regras consistem de uma coleção de regras do tipo "se ... então ..." para a classificação dos dados. A regra antecedente (*se*) é uma expressão feita de conjunções de atributos. A regra consequente (*então*) é uma classificação positiva ou negativa.

Sistemas baseados em regras não têm sido muito usados em SRs. O único trabalho que citamos é Anderson et al. [2003], que implementou um SR de música, baseado em filtragem colaborativa, que melhora o seu desempenho através da aplicação de um sistema de regras sobre as recomendações colaborativas.

2.4.2.3 Classificadores Bayesianos

Consistem de modelos probabilísticos baseado na definição de probabilidade condicional e no teorema de Bayes. O teorema de Bayes utiliza probabilidade para representar a incerteza sobre as relações aprendidas a partir dos dados. Nestes, o conceito de antecedentes é importante, pois eles representam conhecimento a priori sobre as relações.

Os autores Miyahara & Pazzani [2000] modelam o problema de recomendação como o de determinar se um usuário gosta e não gosta de certo item. Os atributos do modelo são avaliações feitas anteriormente, selecionadas (a) após transformação, como um passo do pré-processamento ou (b) se correspondem a dados que os usuários avaliaram em comum. Já Breese et al. [1998] apresenta uma rede Bayesiana onde cada nó corresponde a um item. Os estados de cada nó correspondem a possíveis valores de voto para cada item. Na rede, cada item terá um conjunto de itens pais, que são os seus melhores previsores. As tabelas de probabilidades condicionais são representadas por árvores de decisão, cujos nos correspondem a informação de contexto e as folhas ao voto para cada item. Os autores reportaram melhores resultados para este modelo do que para várias implementações de vizinhos mais próximos sobre um conjunto de dados variado.

2.4.2.4 Redes Neurais Artificiais

Uma Rede Neural Artificial (ANN - *Artificial Neural Network*) Zurada [1992] é um conjunto de nós interconectados com pesos associados, inspirado no cérebro humano. Nós, em uma ANN, são chamados de neurônios. Estas unidades funcionais são compostas em redes que são capazes de aprender um problema de classificação após receberem treinamento com dados.

Christakou & Stafylopatis [2005], usaram ANNs pra implementar um SR de conteúdo, parte de um sistema híbrido, que eles propuseram. O recomendador baseado em conteúdo proposto usou três redes neurais para cada usuário, cada uma delas correspondendo a uma das características a seguir: tipos de filmes, participantes (atores, escritores, roteiristas) e sinopses.

2.4.2.5 Support Vector Machines

Suport Vector Machine (SVM) Cristianini & Shawe-Taylor [2000] são métodos que tratam o problema de classificação como um problema geométrico de separação linear. Em particular SVM procura um hiperplano que separe os dados de forma a maximizar a margem de separação. A maximização feita pelo SVM possibilita que menos itens

desconhecidos possam não ser classificados no futuro.

Xia et al. [2006] também abordam o problema de esparsidade usando SVM. Eles abordam o problema por estimar repetidamente notas em falta. Primeiro, inicializam as notas em falta com valores padrão e, a partir do novo conjunto, constroem classificadores para reestimar as notas.

2.5 Considerações Finais

Neste capítulo foram apresentados conceitos básicos para maior compreensão da pesquisa desenvolvida, bem como um levantamento bibliográfico.

Dentre os trabalhos relacionados, foram levantados artigos de filtragem colaborativa baseada em usuário que exploram diversas características, como notas que usuários dão a itens e suas popularidades, e artigos que incluem característica de confiança. Todos esses trabalhos fazem previsão baseada em fórmulas obtidas heuristicamente. Diferente desses trabalhos, propomos a utilização de técnicas de regressão, Regressão Simples e M5P, e adicionamos novas características para maior caracterização do problema. Acreditamos que métodos heurísticos podem não capturar todas as informações necessárias para se fazer uma boa previsão, por isso o uso de métodos de aprendizado de máquina.

Também foram apresentados trabalhos baseados em aprendizado de ordenação, pois se considerarmos a função de previsão de nota uma função de *ranking*, nosso trabalho consiste em aprender uma função *ranking* e, portanto, é uma estratégia de aprendizado de *ranking*. Até onde sabemos, entretanto, nenhum outro trabalho propôs uma ideia similar à nossa, no contexto de recomendação. Todos os trabalhos de LRT aplicados para SRs que encontramos se baseiam em listas de itens ordenados para realizar o aprendizado, enquanto nosso método aprende a partir de fatores usados em funções de regressão usada por SRs.

Por fim, investigamos se a nossa abordagem foi trabalhada em outro contexto de aprendizado de máquina. Como podemos observar, os métodos de aprendizado de máquina buscam, em sua maioria, melhorias para problemas encontrados em classificação, como dados esparsos. Devido a sua natureza, os métodos abordados necessitam de muita informação para detectar padrões nos dados para a construção do modelo. Em consequência, fazem uso de características de conteúdo enriquecendo perfis de usuários com o intuito de detectar suas preferências. Nosso método não sofre do problema de esparsidade, pois estamos aprendendo funções a partir dos fatores que já são usados pelas formulas heurísticas de regressão.

Capítulo 3

Representação da Nota a ser Prevista

Neste capítulo, apresentamos as várias evidências que exploramos para prever a notas que um usuário daria para um item em um sistema de recomendação. Como observado anteriormente (cf. Seção 2.1.1), o processo de previsão de uma nota em um sistema colaborativo baseado em usuário corresponde a determinar que nota será dada pelo usuário alvo, aquele para o qual se deseja fazer a recomendação, a um certo item com base em notas dadas por usuários similares ao usuário alvo. Logo, três grandes conjuntos de evidências são usadas por tais sistemas no processo de previsão, ou seja, as relacionadas com o (a) usuário alvo, (b) com o item a ser avaliado e (c) com os usuários mais similares ao alvo.

Nas próximas seções descrevemos todos as características relacionadas com esses três conjuntos de evidências, aos quais nos referimos como usuário alvo, item e vizinhança.

3.1 Características do Usuário Alvo

O usuário alvo é o usuário para o qual se deseja realizar uma recomendação. Em uma abordagem de previsão de notas, o processo de recomendação consiste em determinar que nota este usuário daria a um item. Ao observar as Equações 2.2 e 2.5, concluímos que sistemas tradicionais de recomendação colaborativa consideram que, das informações do usuário alvo, a única que deve concorrer para a previsão é o seu grau de rigor, capturado como a nota média que ele forneceu a outros itens. Neste trabalho, além da nota média, estudamos outros atributos específicos do usuário alvo, descritos nas seções a seguir.

3.1.1 Nota média do usuário alvo u ($\overline{r}_u$)

Média das notas atribuídas por u, conforme Equação 3.1.

$$\overline{r}_u = \frac{\sum_{i \in \mathcal{I}_u} r_{iu}}{|\mathcal{I}_u|} \tag{3.1}$$

onde $\mathcal{I}_u$ é o conjunto dos itens avaliados pelo usuário u e r_{iu} é a nota atribuída pelo usuário u ao item i. A nota média do usuário captura o viés de preferência do usuário alvo, ou seja, o seu grau de rigor. A intuição por trás desta caraterística é que nem todos os usuários são igualmente rigorosos. Assim, em um sistema de recomendação de filmes, por exemplo, há usuários que atribuem nota máxima a quaisquer dos filmes que consideram estar entre seus preferidos, enquanto outros usuários dão esta nota a apenas um subconjunto dos seus preferidos por considerarem que a nota máxima deve ser reservada a um grupo seleto de filmes.

A Figura 3.1 mostra a distribuição das notas médias para a coleção Movielens (cf. Seção 4.1). Observe que a maioria das médias está entre 3 e 4 e muito poucos usuários concedem predominantemente notas muito baixas ou muito altas.

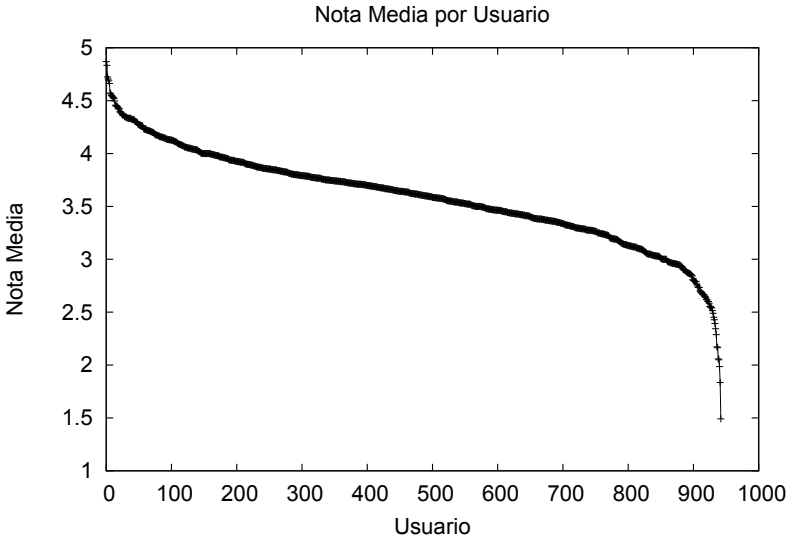


Figura 3.1: Distribuição da nota média dos usuários na coleção.

3.1.2 Desvio padrão da nota média do usuário alvo u (σ_{r_u})

Desvio da média das notas atribuídas por u, conforme Equação 3.2.

$$\sigma_{r_u} = \sqrt{\frac{1}{|\mathcal{I}_u| - 1} \sum_{i \in \mathcal{I}_u} (r_{iu} - \overline{r}_u)^2} \tag{3.2}$$

onde $\mathcal{I}_u$, r_{iu} e $\overline{r}_u$ são descritos na Seção 3.1.1. O desvio da nota média do usuário captura a variação das notas fornecidas pelo usuário. A intuição por trás desta característica é que nem todos os usuários dão notas para itens que cobrem todos os seus tipos de preferência. De fato, é bem conhecido que muitos usuários tendem a atribuir notas apenas para itens que gostam e para itens que rejeitam fortemente.

A Figura 3.2 mostra a distribuição de desvios observados na coleção Movielens (cf. Seção 4.1). Os desvios se concentram entre 0,8 e 1,2.

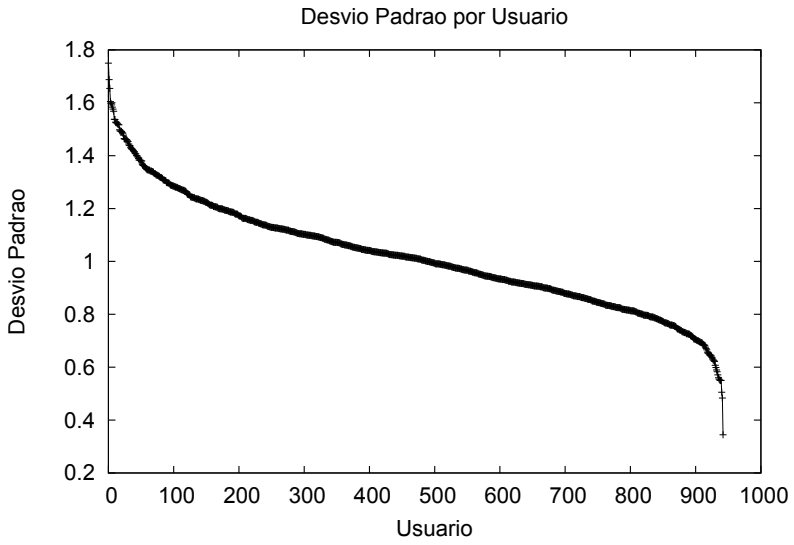


Figura 3.2: Desvio padrão dos usuários na coleção.

3.1.3 Entropia do usuário alvo u (h_u)

Entropia associada às notas atribuídas pelo usuário u, conforme Equação 3.3.

$$h_u = -\sum_{r \in \mathcal{R}} \frac{r}{|\mathcal{I}_u|} \log \frac{r}{|\mathcal{I}_u|} \tag{3.3}$$

onde $\mathcal{R}$ é o conjunto das notas que podem ser atribuídas. Como σ_{r_u}, a entropia das notas médias do usuário captura a variação das notas fornecidas pelo usuário. Ao contrário do desvio padrão, o valor relativo da nota não afeta o cálculo da entropia.

A Figura 3.3 mostra a distribuição de entropias observadas na coleção Movielens (cf. Seção 4.1). Observe que as entropias se concentram entre 1,0 e 1,4.

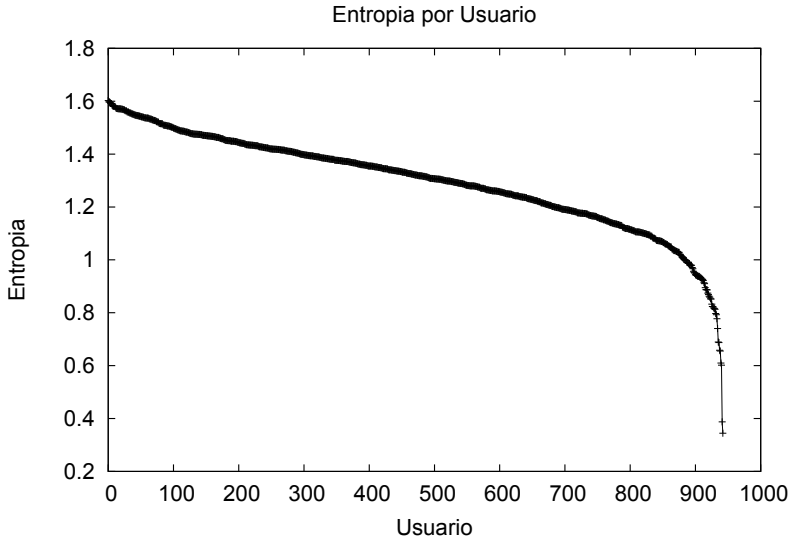


Figura 3.3: Entropia dos usuários na coleção.

3.1.4 Popularidade do usuário alvo u (p_u)

Métrica associada ao número de itens que usuário alvo avaliou, dado por $\log |\mathcal{I}_u|$, onde $\mathcal{I}_u$ é descrito na Seção 3.1.1. A popularidade do usuário alvo está relacionada ao nível de participação dele no sistema de recomendação. A popularidade também ajuda a estimar o quanto as demais métricas descritivas, relacionadas a este usuário, são confiáveis. Assim, o nível de rigor ou a variação com que o usuário fornece notas é provavelmente mais confiável quando obtida para um usuário popular que quando obtida para um usuário que pouco usa o sistema.

A Figura 3.4 mostra a distribuição de popularidade dos usuários para a coleção Movielens (cf. Seção 4.1).

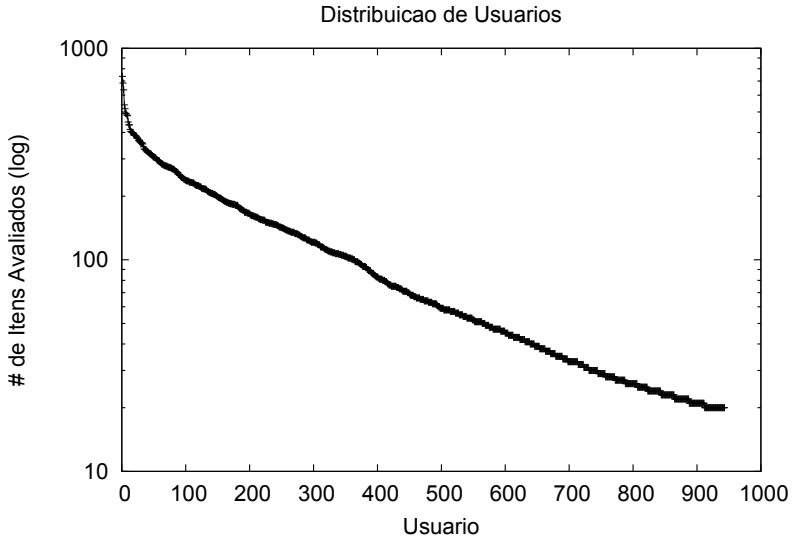


Figura 3.4: Distribuição de popularidade dos usuários na coleção.

3.2 Características do Item

Das Equações 2.2 e 2.5, observamos que sistemas tradicionais de recomendação colaborativa consideram que, das informações do item cuja nota deve ser prevista, tanto a sua popularidade quanto viés de preferência são importantes. Neste trabalho, além destes atributos, estudamos outros específicos do item, descritos nas seções a seguir.

3.2.1 Nota média do item ($\overline{r}_i$)

Média das notas atribuídas ao item i, conforme Equação 3.4.

$$\overline{r}_i = \frac{\sum_{u \in \mathcal{U}_i} r_{iu}}{|\mathcal{U}_i|} \tag{3.4}$$

onde $\mathcal{U}_i$ é o conjunto dos usuários que avaliaram o item i e r_{iu} é a nota atribuída pelo usuário u ao item i. A nota média do item captura a visão que os usuários têm sobre ele, em geral. A intuição por trás desta caraterística é que certos itens são em geral mais bem apreciados que outros. Por exemplo, no domínio de filmes, há filmes que recebem boas notas de quase todos os usuários, enquanto outros não.

A Figura 3.5 mostra a distribuição das notas médias dos itens para a coleção Movielens (cf. Seção 4.1). Podemos observar que poucos itens na coleção possuem notas médias próximas a 5 e que a maioria possui notas médias entre 2,5 e 3,5.

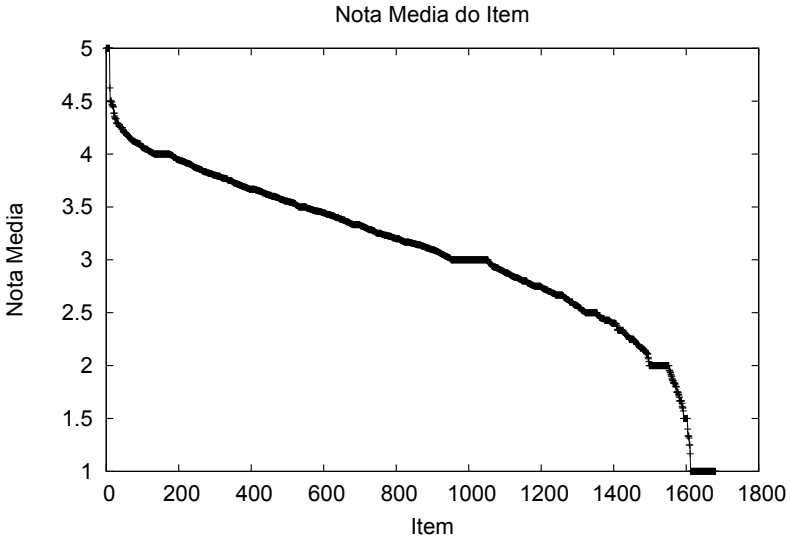


Figura 3.5: Distribuição da nota média dos itens na coleção.

3.2.2 Desvio padrão da nota média do item i (σ_{r_i})

Desvio da média das notas atribuídas para i, conforme Equação 3.5.

$$\sigma_{r_i} = \sqrt{\frac{1}{|\mathcal{U}_i| - 1} \sum_{u \in \mathcal{U}_i} (r_{iu} - \overline{r}_i)^2} \tag{3.5}$$

onde $\mathcal{U}_i$, r_{iu} e $\overline{r}_i$ são descritos na Seção 3.2.1. O desvio da nota média do item captura a variação das notas fornecidas para o item. A intuição por trás desta caraterística é que certos itens podem ter distribuição de notas características como avaliações majoritariamente positivas ou negativas.

A Figura 3.6 mostra a distribuição de desvios das médias dos itens observados na coleção Movielens (cf. Seção 4.1). Podemos ver que o desvio padrão é menor que 1 na maior parte dos itens, o que significa que a média de notas não está distante das notas que realmente são atribuídas ao item.

3.2.3 Entropia do item i (h_i)

Entropia associada às notas atribuídas ao item i, conforme Equação 3.6.

$$h_i = -\sum_{r \in \mathcal{R}} \frac{r}{|\mathcal{U}_i|} \log \frac{r}{|\mathcal{U}_i|} \tag{3.6}$$

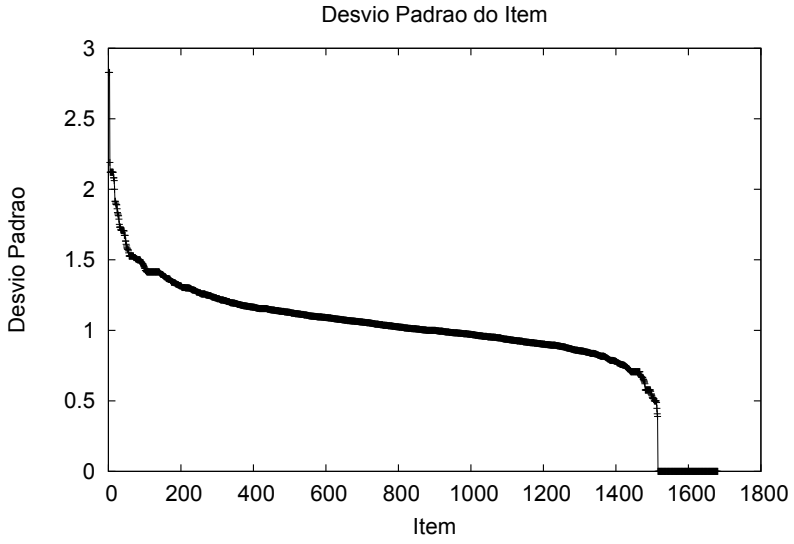


Figura 3.6: Desvio padrão dos itens na coleção.

onde $\mathcal{R}$ é o conjunto das notas que podem ser atribuídas. Como σ_{r_i}, a entropia das notas médias do item captura a variação das notas fornecidas para o item, com a peculiaridade de que o valor relativo da nota não afeta a entropia.

A Figura 3.7 mostra a distribuição de entropias das notas dos itens observadas na coleção Movielens (cf. Seção 4.1). Observe que as entropias estão no intervalo entre 1,0 e 1,4.

3.2.4 Popularidade do item i (p_i)

Métrica associada ao número de usuários que avaliaram o item i, dada por $\log |\mathcal{U}_i|$, onde $\mathcal{U}_i$ é descrito na Seção 3.2.1. A popularidade do item está relacionada à visibilidade do item junto aos usuários. Ela também é útil como indicação de quanto as demais métricas descritivas, relacionadas a este item, são confiáveis.

A Figura 3.8 mostra a distribuição de popularidade dos itens para a coleção Movielens (cf. Seção 4.1).

3.3 Características da Vizinhança

Como observado nas Equações 2.2 e 2.5, métodos tradicionais de recomendação colaborativa consideram que a informação fundamental para a previsão de uma nota são

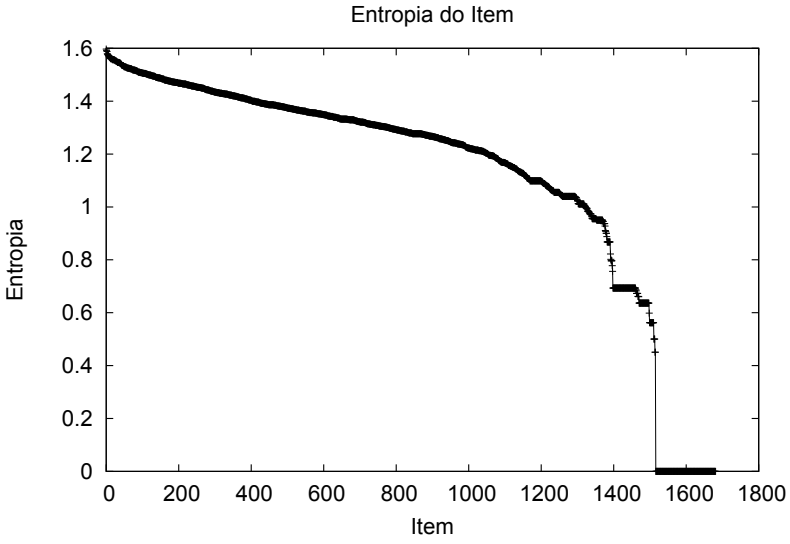


Figura 3.7: Entropia dos itens na coleção.

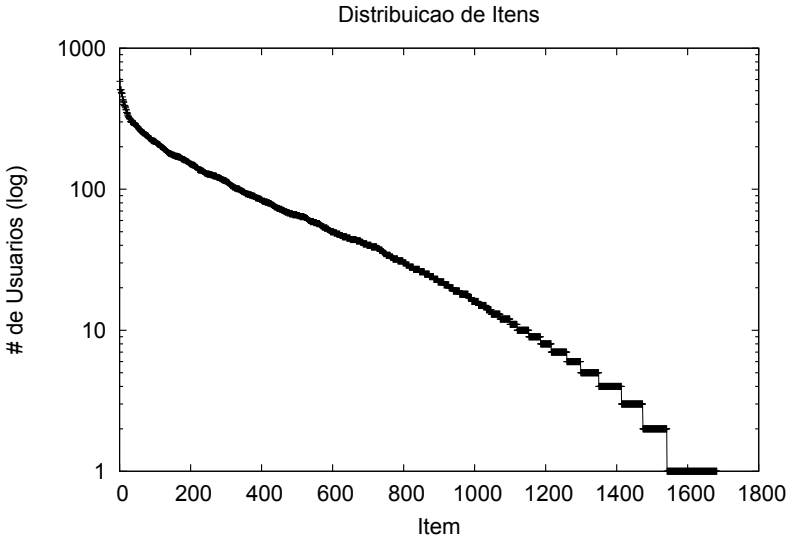


Figura 3.8: Distribuição de popularidade dos itens na coleção.

as notas fornecidas por usuários semelhantes ao usuário alvo, ou seja, usuários em sua vizinhança de similaridade. Tais notas podem ser corrigidas pela confiança do usuário alvo no vizinho e pelo viés de preferência do vizinho.

Nas próximas seções, descrevemos as características usadas, iniciando por aquelas

de natureza global e, então, finalizando nas que descrevem os vizinhos individualmente.

3.3.1 Tamanho da vizinhança ($|\mathcal{V}_{ui}|$)

Seja $\mathcal{V}_{ui}$ o conjunto de usuários na vizinhança do usuário u que atribuíram nota ao item i. O tamanho da vizinhança é dado por $|\mathcal{V}_{ui}|$. Esta métrica é útil uma vez que vizinhanças maiores podem implicar em estatísticas mais confiáveis. A variação do tamanho da vizinhança, contudo, só ocorre quando a inclusão de um vizinho é determinada por um limiar como, por exemplo, sua confiança. Por exemplo, a Figura 3.9 apresenta a distribuição de tamanhos de vizinhança para o método FbC-Donovan no Movielens, considerando que os vizinhos tenham grau de confiança maior ou igual a 0,67 para o usuário alvo. Como podemos notar, a variância do tamanho da vizinhança considerada para diferentes usuários alvo é bastante alta.

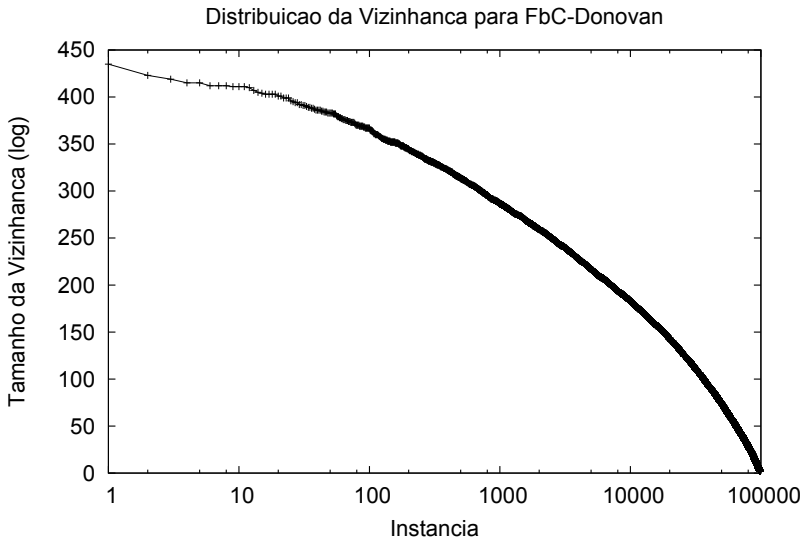


Figura 3.9: Distribuição dos tamanhos das vizinhanças resultantes do método FbC-Donovan.

3.3.2 Nota média da vizinhança ($\overline{r}_{\mathcal{V}_{ui}}$) e desvio da nota ($\sigma_{\overline{r}_{\mathcal{V}_{ui}}}$)

Média ponderada das notas atribuídas por todos os vizinhos, calculada conforme Equação 3.7.

$$\overline{r}_{\mathcal{V}_{ui}} = \overline{r}_u + \frac{\sum_{v \in \mathcal{V}_{ui}} S_{uv}(r_{vi} - \overline{r}_v)}{\sum_{v \in \mathcal{V}_{ui}} S_{uv}} \tag{3.7}$$

onde $\mathcal{V}_{ui}$ é o conjunto de vizinhos de u que avaliaram i, S_{uv} corresponde à similaridade entre u e v, r_{vi} é a nota atribuída pelo vizinho v ao item i e $\overline{r}_j$ é a nota média dada pelo usuário j. A nota média da vizinhança corresponde à Equação 2.2.

Além da média, com o intuito de capturar o grau de variância desta média, também incluímos como característica o desvio padrão, calculado conforme Equação 3.8.

$$\sigma_{\overline{r}_{\mathcal{V}_{ui}}} = \sqrt{\frac{1}{|\mathcal{V}_{ui}| - 1} \sum_{v \in \mathcal{V}_{ui}} (r_{vi} - \overline{r}_{\mathcal{V}_{ui}})^2} \tag{3.8}$$

A intuição por trás desta caraterística é que a contribuição dos vários vizinhos nas vizinhanças pode não ser uniforme. Isto sugere que algumas médias podem ser mais confiáveis que outras.

3.3.3 Similaridade média da vizinhança ($\overline{S}_{\mathcal{V}_{ui}}$) e desvio da similaridade ($\sigma_{\overline{S}_{\mathcal{V}_{ui}}}$)

Média ponderada das similaridades dos vizinhos, calculada conforme Equação 3.9.

$$\overline{S}_{\mathcal{V}_{ui}} = \frac{1}{\mathcal{V}_{ui}} \sum_{v \in \mathcal{V}_{ui}} S_{uv} \tag{3.9}$$

onde $\mathcal{V}_{ui}$ e S_{uv} são dadas na Seção 3.3.2. A similaridade da vizinhança captura o quão similar é esta vizinhaça como um todo, independente das notas dadas. Com intuito de capturar o grau de variância destas similaridades, também incluimos como característica o desvio padrão, calculado conforme Equação 3.10.

$$\sigma_{\overline{S}_{\mathcal{V}_{ui}}} = \sqrt{\frac{1}{|\mathcal{V}_{ui}| - 1} \sum_{v \in \mathcal{V}_{ui}} (\overline{S}_{\mathcal{V}_{ui}})} \tag{3.10}$$

Como antes, essa caraterística captura o quão uniforme são as similaridades dos vizinhos, o que pode ser um indício de quão confiáveis são previsões feitas com base nessa vizinhança.

3.3.4 Características dos vizinhos

Além de descrever a vizinhança usando estatísticas globais, como as descritas em seções anteriores, iremos estudar o impacto de estatísticas individuais de cada vizinho no processo de aprendizado.

Note, contudo, que para grandes vizinhanças, o número de características a serem usadas na representação se tornaria proibitivamente grande. Para contornar esse

problema, nós discretizamos a vizinhança de forma que ela seja constituída por grupos de n usuários em lugar dos k usuários individuais. Assim, para as métricas descritas nesta seção, chamamos de *vizinho* um grupo de usuários que se encontra dentro de uma certa faixa de distância do usuário alvo, de acordo com uma métrica de similaridade. Desta forma, uma métrica de um vizinho se refere a média daquela métrica tomada sobre todos os usuários que constituem o vizinho. Para ilustrar essa idéia, a Figura 3.10 exibe a vizinhança do usuário u_1, constituída por usuários u_2 a u_{11}. Neste exemplo, a vizinhança $\mathcal{V}_{u_1 i}$ é dividida em três grupos de usuários, $v_1 = \{u_2, u_3, u_4\}$, $v_2 = \{u_5, u_6, u_7\}$ e $v_3 = \{u_8, u_9, u_{10}, u_{11}\}$. Neste exemplo, a similaridade do vizinho v_1 para u_1 é dada como a média das similaridades de u_2, u_3 e u_4 para u_1.

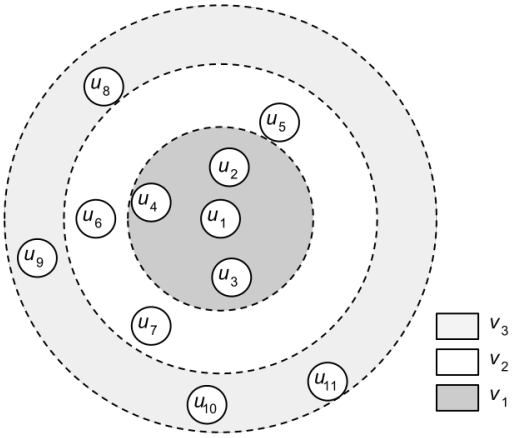


Figura 3.10: Usuário u_1 e sua vizinhança, dividida em três conjuntos de usuários, v_1, v_2 e v_3.

Assim, para cada vizinho, as seguintes métricas são usadas na representação da nota a ser prevista, onde $\mathcal{I}_v$ corresponde ao conjunto de itens que o vizinho v avaliou:

- $\mathbf{s}_{uv}$: similaridade entre o vizinho v e usuário alvo, calculada através da correlação de Pearson descrita na Seção 2.1.1.

- $\mathbf{r}_{vi}$: nota que o vizinho v forneceu ao item i.

- $\mathbf{p}_v$: popularidade do vizinho v, calculada como $\log |\mathcal{I}_v|$.

- $\overline{\mathbf{r}}_v$: nota média dos vizinhos v, obtida como $\overline{r}_v = \frac{\sum_{i \in \mathcal{I}_v} r_{vi}}{|\mathcal{I}_v|}$.

3.4 Considerações Finais

Neste capítulo foram apresentadas estatísticas e caracaterizações dos dados utilizados nesta dissertação. Esses, foram divididos em três grandes grupos: características de usuários, de item e vizinhança. As características nota média do usuário alvo u ($\overline{r}_u$), popularidade de u (p_u), nota média do item i ($\overline{r}_i$), popularidade de i (p_i) e tamanho da vizinhança ($|\mathcal{V}_{ui}|$) foram extraídas do processamento do Mahout ao realizar as recomendações de acordo com os *baselines*. A partir dessas realizamos o processamento das demais utilizando a linguagem Python[1]. As características processadas nessa segunda fase foram o desvio padrão da nota média do usuário u (σ_{r_u}), entropia do usuário u (h_u), desvio padrão da nota média do item i (σ_{r_i}),entropia do item i (h_i), nota média da vizinhança ($\overline{r}_{\mathcal{V}_{ui}}$) e seu desvio padrão ($\sigma_{\overline{r}_{\mathcal{V}_{ui}}}$), similaridade média da vizinhança ($\overline{\mathcal{S}}_{\mathcal{V}_{ui}}$) e seu desvio ($\sigma_{\overline{\mathcal{S}}_{\mathcal{V}_{ui}}}$) e por fim, as características de vizinhos similaridade entre vizinho v e usuário alvo u ($\mathbf{s}_{uv}$), nota que v forneceu ao item i ($\mathbf{r}_{vi}$), popularidade do vizinho v ($\mathbf{p}_v$) e nota média de v ($\overline{\mathbf{r}}_v$). Estas são em sua forma primária obtidas a partir do Mahout também, porém como foi explanado na seção anterior (cf. Seção 3.3.4), as características de vizinho foram processadas de outra forma neste trabalho, portanto diz respeito a segunda forma de obtenção de características.

[1] www.python.org

Capítulo 4

Experimentos

Neste capítulo, avaliamos o uso de métodos de regressão com o conjunto de atributos usados em métricas tradicionais de filtragem colaborativa baseada em usuário, com uso ou não de informação de reputação (confiança entre usuários). Os atributos usados são aqueles apresentados no Capítulo 3. Mais especificamente, ao longo deste capítulo apresentamos a coleção usada na avaliação, os experimentos realizados e os resultados obtidos.

4.1 Coleção de Referência Movielens

Para avaliar nossa hipótese neste trabalho, usamos a coleção de referência Movielens[1]. Esta é uma coleção comumente usada para avaliar estratégias de recomendação de filmes. Foi obtida pelo GroupLens Research Project, da Universidade de Minnesota, do *site* MovieLens[2] durante sete meses, entre Setembro de 1997 e Abril de 1998.

Contém 100.000 notas entre 1 (pior) a 5 (melhor), atribuídos por 943 usuários a 1682 filmes, onde cada usuário avaliou pelo memos 20 filmes. Filmes e usuários são descritos por identificadores aleatórios começando em um. Os filmes correspondem aos gêneros ação, aventura, animação, infantil, comédia, crime, documentário, drama, fantasia, filme *noir*, horror, musical, mistério, romance, fição científica, suspense, guerra e *western*.

Na Figura 4.1, temos as distribuições de notas na coleção Movielens. Observe na figura que a nota média encontra-se em torno de 4, com a maioria das notas se concentrando entre 3 a 5, o que indica que os usuários geralmente avaliam filmes que eles

[1]http://www.movielens.org
[2]http://movielens.umn.edu

gostam. Este fenômeno é comumente observado em sistemas de recomendação [Ricci et al., 2011].

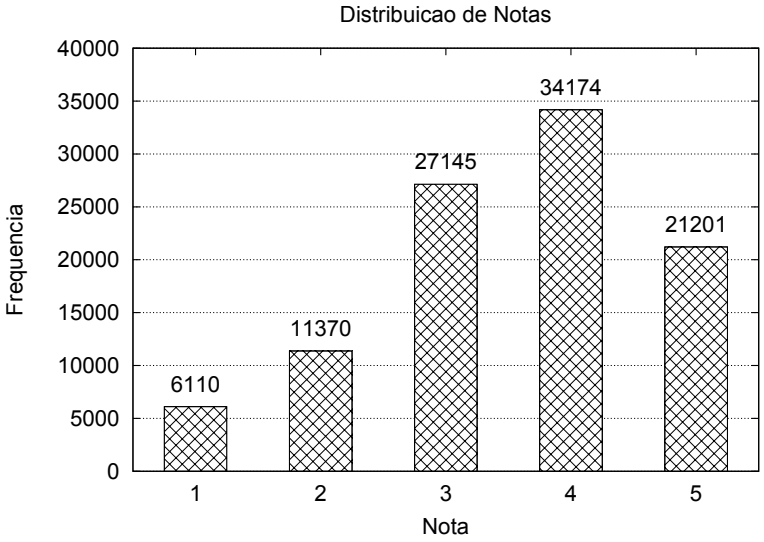


Figura 4.1: Distribuição de notas na coleção.

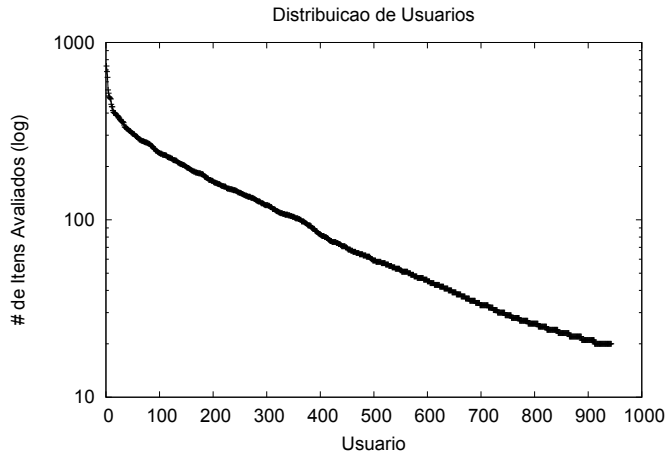
As Figuras 4.2 (a) e (b) mostram as distribuições de usuários e itens na coleção Movielens. As duas distribuições são de cauda longa, ou seja, (a) a maioria dos usuários avalia poucos filmes (neste caso, a cauda é truncada, já que apenas usuários que avaliaram pelo menos 20 filmes aparecem na coleção) e (b) a maioria dos filmes foi avaliada por poucos usuários.

Embora simples, esta caracterização da coleção Movielens nos permite afirmar que ela apresenta o comportamento típico de uma coleção de itens a serem recomendados.

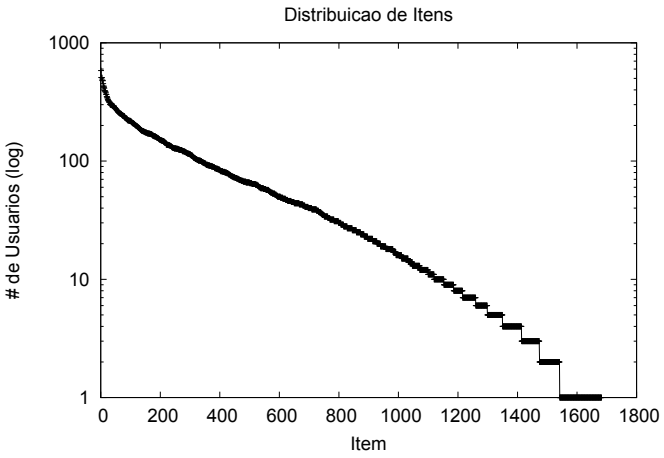
4.2 Metodologia

Nossos experimentos foram realizados com os seguintes objetivos. Primeiro, avaliar a qualidade da previsão com todas as evidências utilizadas pelos *baselines* FCU-Resnick e FbC-Donovan (que inclui informação de reputação). Segundo, avaliar o impacto de evidências adicionais, derivadas diretamente das evidências usadas pelos *baselines*. E, por fim, analisar o impacto de cada evidência para o método de melhor desempenho entre os estudados.

Para avaliar o desempenho dos métodos estudados, utilizamos a métrica RMSE (cf. Seção 2.3.2). Todos os experimentos foram realizados utilizando o método de vali-



(a)



(b)

Figura 4.2: Distribuição de usuários na coleção (a); Distribuição de itens (filmes) na coleção (b).

dação cruzada (ver Seção 2.3.1) de 5-*folds*, considerando a divisão padrão recomendada para a coleção Movielens. Assim, os resultados reportados nesta dissertação referem-se à média dos 5-*folds*. Todas as comparações feitas neste trabalho foram verificadas quanto à sua significância estatística utilizando o teste de Wilcoxon (cf. Seção 2.3.1). Com exceção dos casos onde é explicitamente informado o contrário, todas as comparações reportadas neste trabalho apresentam grau de confiança maior que 95%.

4.3 Resultados

Nesta seção apresentamos os resultados obtidos, ou seja, (a) a comparação entre os métodos de regressão linear e M5P com os métodos FCU-Resnick e FbC-Donovan; e (b) o estudo das características usadas.

4.3.1 Comparação entre métodos tradicionais e regressor linear e M5P

A Tabela 4.1 apresenta os valores de RMSE obtidos para o método *baseline* FCU-Resnick e para os métodos de regressão linear e M5P. No caso do método FCU-Resnick, foram testados valores de k entre 2 e 90. Destes, alguns são exibidos na tabela, em particular, $k = 10$ (valor padrão do Mahout), $k = 20$ (valor muito usado na literatura de recomendação de filmes) e $k = 90$ (valor que corresponde ao tamanho médio da vizinhança para a melhor configuração do método FbC-Donovan, como veremos adiante).

Note que as características usadas pelos métodos de regressão linear e M5P foram as usadas em FCU-Resnick, ou seja, a nota média do usuário ($\overline{r}_u$), a similaridade entre vizinho v e usuário ($\mathbf{s}_{uv}$), a nota que o vizinho v forneceu ao item i ($\mathbf{r}_{vi}$), a nota média do vizinho v ($\overline{\mathbf{r}}_v$), a popularidade do usuário alvo u (p_u), a popularidade do item i (p_i) e a popularidade do vizinho v ($\mathbf{p}_v$).

Métodos	**k=10**		**k=20**		**k=90**	
	Erro	**Ganho**	**Erro**	**Ganho**	**Erro**	**Ganho**
FCU-Resnick	1,1867	-	1,1534	-	1,0712	-
Regressão Linear	1,0157	14,41%	1,0120	12,26%	1,0015	6,51%
M5P	1,0162	14,37%	1,0120	12,26%	0,9996	6,68%

Tabela 4.1: Resultados para *baseline* FCU-Resnick, Regressão Linear e M5P usando apenas características usadas por FCU-Resnick.

Como podemos observar, tanto a regressão linear quanto o M5P apresentaram erros de previsão menores que o FCU-Resnick, com ganho mínimo de 6% para o maior valor de k (ganho estatísticamente significativo). Não houve diferença estatística entre o regressor linear e o M5P neste primeiro conjunto de experimento. Também notamos que o ganho é menor à medida que o valor de k é maior, o que pode ser observado mais claramento na Figura 4.3.

Um fator que pode contribuir para este resultado é o fato que quanto maior o k, mais esparsos os dados se tornam, o que dificulta o aprendizado dos regressores.

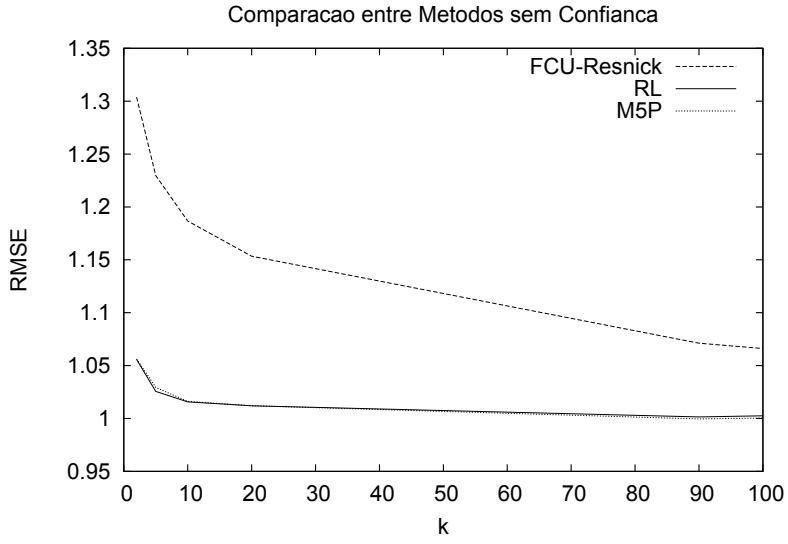


Figura 4.3: Comparação entre o FCU-Resnick e os métodos de regressão Regressão Linear (RL) e M5P.

Isto ocorre porque, para alguns usuários, o número de vizinhos similares k_{sim} é menor que k. Para estes casos, os valores para o i-ésimo vizinho mais próximo, $i \leq k_{sim}$, são preenchidos com zero.

Em um segundo conjunto de experimentos, avaliamos o impacto da inclusão de informação de reputação. Neste caso, avaliamos os métodos propostos por O'Donovan & Smyth [2005]. Destes experimentos, observamos que o método de filtro descrito na Seção 2.1.2.2 (que chamamos FbC-Donovan) obteve os melhores resultados para a coleção Movielens[3]. Em particular, o melhor resultado foi obtido ao usar os parâmetros limiar de confiança c_{min} igual a 0,67.

Para os métodos de regressão, utilizamos as mesmas evidências do experimento anterior, porém, calculadas sobre uma vizinhança onde a confiança entre qualquer vizinho e o usuário alvo é maior ou igual a c_{min}, assim como em FbC-Donovan. Note que nesta situação, a vizinhança selecionada tem, em média, 90 vizinhos. Contudo, em alguns casos, a vizinhança pode ter aproximadamente 500 vizinhos. Este número torna proibitiva a aplicação de um método de aprendizado de máquina, uma vez que implica em uma dimensão muito alta e dados muito esparsos. Para evitar este problema, as métricas dos vizinhos foram tomadas em grupos de vizinhança em um processo de discretização apresentado na Seção 3.3.4. Como muitos tamanhos de grupos de vizinhos

[3]Estes experimentos foram feitos no contexto de um trabalho de iniciação científica.

(n) são possíveis, experimentamos com diversos valores para n, em particular, 10, 20, 30, 50, 150 e 200.

Os resultados obtidos para esse experimento são sumarizados na Figura 4.4. Nela, são exibidos os resultados obtidos para o (a) Regressor Linear, (b) o M5P e (c) o método FCU-Resnick ($k = 90$, o tamanho médio da vizinhança para FbC-Donovan), considerando diferentes tamanhos de grupo de vizinhos (n), além (d) do método FbC-Donovan [O'Donovan & Smyth, 2006].

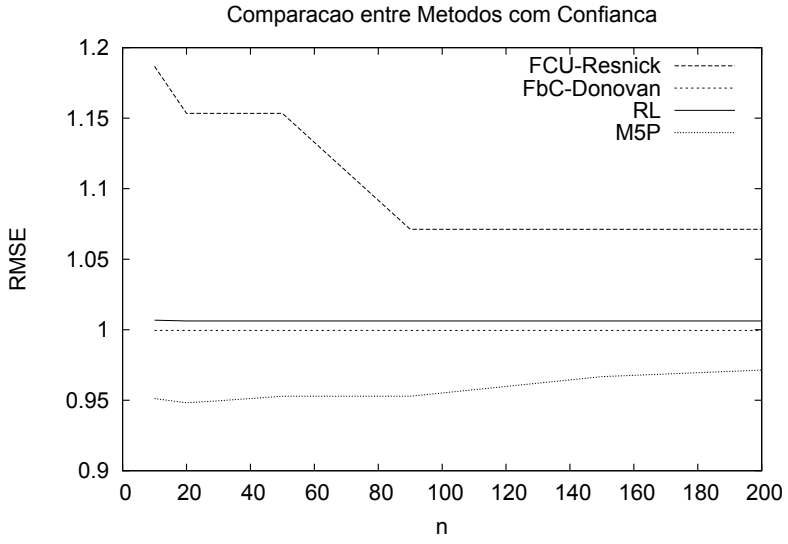


Figura 4.4: Comparação entre os métodos de regressão Regressão Linear (RL) e M5P e os *baselines* FCU-Resnick e FcB-Donovan.

Podemos observar nessa figura que embora tenha desempenho significativamente superior ao FCU-Resnick, a Regressão Linear apenas empata com o método FbC-Donovan. Isso pode implicar que a Regressão Linear não foi capaz de aprender nada que o método FbC-Donovan já não explore. O método M5P, ao contrário, foi capaz de melhorar em até 5% o melhor resultado de FbC-Donovan (para $n = 20$). Além disso, também podemos observar que quanto maior o valor de n, menor é o ganho do M5P em relação ao FbC-Donovan. Isto pode estar relacionado com a maior heterogeneidade dos conjuntos à medida que aumentamos o valor de n. Note que o mesmo é observado para o FCU-Resnick que, surpreendentemente, também melhora o seu desempenho quando é calculado para vizinhanças formadas por grupos de vizinhos.

4.3.2 Estudo de Características

Nesta seção apresentamos o estudo do desempenho das características com o propósito de (a) determinar se os regressores podem melhorar seu desempenho ao usarem características derivadas diretamente das usadas pelos *baselines* e (b) determinar qual a contribuição de cada característica para a previsão final.

Em um primeiro experimento, nós utilizamos nossos melhores regressores anteriores, ou seja, Regressão Linear e M5P, usando vizinhança selecionada conforme confiança. No caso do M5P, usamos $n = 50$, que foi o nosso melhor resultado quando consideramos todas as características. As características usadas foram todas as exploradas anteriomente (a popularidade do usuário alvo u (p_u), a nota média do usuário alvo ($\overline{r}_u$), a popularidade do item i (p_i), a similaridade entre vizinho v e usuário ($\mathbf{s}_{uv}$), a nota que vizinho v forneceu ao item i ($\mathbf{r}_{vi}$), a popularidade do vizinho v ($\mathbf{p}_v$) e a nota média do vizinho v ($\overline{\mathbf{r}}_v$)), bem como algumas adicionais. Estas são a nota média do item i ($\overline{r}_i$), o desvio padrão da $\overline{r}_u$ (σ_{r_u}), a entropia do usuário alvo u (h_u), o desvio padrão da $\overline{r}_i$ (σ_{r_i}), a entropia do item i (h_i), o tamanho da vizinhança ($|\mathcal{V}_{ui}|$), a nota média da vizinhança ($\overline{r}_{\mathcal{V}_{ui}}$), o desvio da $\overline{r}_{\mathcal{V}_{ui}}$ ($\sigma_{\overline{r}_{\mathcal{V}_{ui}}}$), a similaridade média da vizinhança ($\overline{S}_{\mathcal{V}_{ui}}$) e o desvio da $\overline{S}_{\mathcal{V}_{ui}}$ ($\sigma_{\overline{S}_{\mathcal{V}_{ui}}}$).

Os resultados desse experimento podem ser observados na Tabela 4.2. Nesta tabela, comparamos os métodos de Regressão Linear (RL) e M5P usando ou não todas as características propostas neste trabalho. No caso da RL, usamos $k = 90$ e ganhos são dados em referência ao FCU-Resnick. No caso do M5P, usamos $n = 50$ e ganhos são dados em referência ao FbC-Donovan. Como podemos observar, o uso de características adcionais melhorou o desempenho do M5P, mas não do regressor linear. No caso do M5P, o ganho final ficou na ordem de 8% sobre nosso melhor baseline, o FbC-Donovan.

Métodos	**RMSE**	**Ganho**
FCU-Resnick	1,0712	-
RL apenas com características de FCU-Resnick	1,0015	6,51%
RL com todas características	1,0062	6,06%
FbC-Donovan	0,9995	-
M5P apenas com características de FCU-Resnick	0,9482	5,13%
M5P com todas características	0,9205	7,91%

Tabela 4.2: Resultados Regressão Linear (RL) e M5P usando todas características propostas neste trabalho. Em ambos os casos, usamos seleção baseada em filtragem de confiança. No caso da RL, usamos $k = 90$ e ganhos são dados em referência ao FCU-Resnick. No caso do M5P, usamos $n = 50$ e ganhos são dados em referência ao FbC-Donovan.

Em um segundo experimento, avaliamos o impacto das várias características estudadas. Neste experimento, usamos apenas o regressor com o qual obtivemos o melhor desempenho, ou seja, o método M5P com vizinhança selecionada por filtragem de confiança e $n = 50$ vizinhos por grupo. Para avaliar a importância das características, regressões foram realizadas com o M5P tendo (1) cada característica sendo aplicada individualmente e (2) removida do conjunto usado para representar cada nota a ser prevista. No primeiro caso, esperamos observar a importância de cada característica individualmente enquanto no segundo caso, procuramos determinar se ela não é redundante.

A Tabela 4.3 apresenta os resultados obtidos. Dada uma característica, o valor do RMSE obtido para ela tomada individualmente é dado na coluna "Isolada", enquanto o valor do RMSE para o conjunto de característica, excelo ela, é dado na coluna "Excluída". As colunas ganho e perda indicam o ganho de desempenho relacionado com o uso individual da característica e a perda de desempenho relacionada com a sua remoção do conjunto de características. Assim, um atributo é melhor na medida em que resulta em maior ganho quando usado isoladamente e maior perda quando excluído. Perdas e ganhos são calculados em referência ao desepenho do M5P com todas as características (RMSE = 0,9205).

Como podemos observar na Tabela 4.3, as notas ($\overline{r}_u$, $\overline{r}_i$, $\overline{r}_{\mathcal{V}_{ui}}$, $\overline{\mathbf{r}}_v$ e $\mathbf{r}_{vi}$) são as características mais importantes. Elas tanto são as que resultam nos melhores desempenhos isolados quanto nas maiores perdas, quando removidas. De fato, quando usadas apenas estas características na regressão com M5P, obtivemos um RMSE de 0,9281, apenas um pouco menor que o obtido com todas as características. Isso explica os bons resultados das heurísticas tradicionais que são focadas nessas características, de uma forma ou outra. Entre elas, o melhor desempenho foi exatamente do conjunto de notas que os vizinhos forneceram ao item ($\mathbf{r}_{vi}$).

Também notamos que poucas características representaram efetivamente perdas ($|\mathcal{V}_{ui}|$, $\mathbf{s}_{uv}$ e $\mathbf{p}_v$), o que implica que no pior dos casos, a maioria das características ou não contribui ou contribui levemente para o resultado final. Em particular, os piores resultados são relacionados com características dos vizinhos. Em parte, isso se justifica pelo fato de que como estas características representam conjuntos de valores e não valores isolados, elas têm proporcionalmente mais impacto para o aprendizado, seja positiva ($\mathbf{r}_{vi}$) ou negativamente ($\mathbf{s}_{uv}$ e $\mathbf{p}_v$).

Características	Isolada	Ganho	Excluída	Perda
	RMSE	(%)	**RMSE**	(%)
Nota média do usuário alvo u ($\overline{r}_u$)	1,0401	-13,00	0,9277	0,78
Desvio padrão da $\overline{r}_u$ (σ_{r_u})	1,0524	-14,33	0,9207	0,03
Entropia do usuário alvo u (h_u)	1,0478	-13,83	0,9213	0,09
Popularidade do usuário alvo u (p_u)	1,1191	-21,58	0,9207	0,02
Nota média do item ($\overline{r}_i$)	1,0059	-9,28	0,9283	0,85
Desvio padrão da $\overline{r}_i$ (σ_{r_i})	1,0234	-11,18	0,9207	0,02
Entropia do item i (h_i)	1,0233	-11,17	0,9215	0,10
Popularidade do item i (p_i)	1,0843	-17,79	0,9210	0,06
Tamanho da vizinhança ($\lvert\mathcal{V}_{ui}\rvert$)	1,0649	-15,69	0,9204	-0,01
Nota média da vizinhança ($\overline{r}_{\mathcal{V}_{ui}}$)	0,9979	-8,41	0,9227	0,24
Desvio da $\overline{r}_{\mathcal{V}_{ui}}$ ($\sigma_{\overline{r}_{\mathcal{V}_{ui}}}$)	1,0908	-18,50	0,9205	0,00
Similaridade média da vizinhança ($\overline{S}_{\mathcal{V}_{ui}}$)	1,0985	-19,34	0,9215	0,11
Desvio da $\overline{S}_{\mathcal{V}_{ui}}$ ($\sigma_{\overline{S}_{\mathcal{V}_{ui}}}$)	1,0968	-19,15	0,9206	0,01
Similaridade entre vizinho v e usuário ($\mathbf{s}_{uv}$)	1,0699	-16,23	0,9203	-0,02
Nota que vizinho v forneceu ao item i ($\mathbf{r}_{vi}$)	0,9994	-8,57	0,9352	1,60
Popularidade do vizinho v ($\mathbf{p}_v$)	1,0602	-15,18	0,9203	-0,02
Nota média do vizinho v ($\overline{\mathbf{r}}_v$)	1,0206	-10,88	0,9231	0,28

Tabela 4.3: Resultados para o estudo de característica utilizando o M5P com discretização $n = 50$. Perdas e ganhos são calculados em relação ao M5P com todas as características, cujo RMSE = 0,9205. Note que os atributos estão agrupados conforme sua origem: usuário, item, vizinhança e vizinhos.

4.4 Considerações Finais

Nesse capítulo foram apresentados os experimentos realizados, os resultados obtidos e a coleção de referência utilizada. Os experimentos foram feitos com os objetivos de (i) avaliar a qualidade da previsão com as evidências utilizadas pelos *baseline*; (ii) avaliar o impacto das evidências adicionais que propomos; e (iii) analisar o impacto de cada evidência para o método com melhor desempenho entre os estudados.

A partir do experimentos observamos que os métodos de regressão baseados em aprendizado de máquina aqui avaliados, foram superiores aos métodos *baseline*, baseados em fórmulas heurísticas, validando, assim, nossa hipótese inicial. Observamos também, que as evidências que possuem maior influência nos resultados são as características (i) Nota média do usuário alvo u ($\overline{r}_u$); (ii) Nota média do item ($\overline{r}_i$); (iii) Nota média da vizinhança ($\overline{r}_{\mathcal{V}_{ui}}$); (iv) Nota que vizinho v forneceu ao item i ($\mathbf{r}_{vi}$); e (v) Nota média do vizinho v ($\overline{\mathbf{r}}_v$). Detalhes acerca das conclusões alcançadas são apresentados no Capítulo 5.

Capítulo 5

Conclusões e Trabalhos Futuros

Neste capítulo, apresentamos as conclusões do nosso trabalho, incluindo limitações de nossa pesquisa e direções futuras que podem ser exploradas.

5.1 Resultados Obtidos

Neste trabalho, verificamos o uso de métodos de aprendizagem de máquina para o problema previsão de notas em Sistemas de Recomendação, em lugar de fórmulas fechadas. Mais especificamente, observamos o uso dos métodos de regressão Regressão Linear e M5P e também o impacto de informação de reputação em filtragem colaborativa baseada em usuário. Assim, nossos experimentos consideraram dois cenários. No primeiro, vizinhanças são selecionadas pela similaridade entre usuários; no segundo, a vizinhança também leva em conta similaridades, porém é filtrada conforme a reputação dos usuários. No primeiro cenário comparamos a Regressão Linear e o M5P ao método FCU-Resnick. No segundo, realizamos a comparação dos regressores com o método FbC-Donovan.

De forma geral, nossos resultados mostraram que as regressões baseadas em aprendizagem de máquina foram superiores aos métodos tradicionais baseados em equações heurísticas. Mais especificamente, nosso principal resultado mostra que o M5P obteve ganhos significativos sobre o FCU-Resnick quando utiliza a mesma abordagem e ganhos maiores ainda quando utiliza informação de reputação. Nesse caso, ele também superou o principal *baseline*, o FbC-Donovan.

Realizamos, também, um estudo das características utilizadas ao longo de toda a experimentação. Esse estudo nos mostrou que as evidências que representam notas que vizinhos dariam para o item (como um todo ou individualmente) e as notas médias do usuário e do item são as mais importantes. Também notamos que poucas características

pioram o resultado de forma estatísticamente significativa. Estas observações tanto explicam o bom desempenho dos *baselines* (focados em evidências de notas) quanto o fato de que as evidências adicionais melhoram o resultado final, embora com um pequeno ganho.

Por fim, a motivação para esta pesquisa foi sintetizada em quatro perguntas, apresentadas na Seção 1.1, para as quais obtivemos as seguintes respostas:

- *Em lugar de usar fórmulas heurísticas, não seria mais eficaz determinar automaticamente, por meio de uma técnica de aprendizagem de máquina, a melhor combinação das evidências disponíveis de forma a reduzir o erro de previsão?* Os experimentos mostraram que as regressões baseadas em aprendizagem de máquina (Regressão Linear e M5P) apresentaram melhores resultados que métodos tradicionais. Contudo, quando considerado o cenário com informação de confiança, apenas o método M5P foi capaz de superar o melhor *baseline*.

- *Evidências derivadas das originais não poderiam ser mais eficazes em termos de facilitar a previsão das notas?* Em nossos experimentos, os melhores resultados alcançados foram obtidos com as evidências adicionais que propusemos. Contudo, apenas o método M5P tirou proveito dessas evidências, alcançando melhores resultados.

- *Dentre as evidências usadas para realizar a previsão, (sejam as tradicionalmente usadas, sejam as novas derivadas pra este problema) quais são as mais importantes, considerando a sua contribuição para a minimização do erro?* As evidências que obtiveram melhores resultados são as relacionadas com as notas que vizinhos dariam pro item (como um todo ou individualmente) e as notas médias do usuário, do item e dos vizinhos. Contudo, com exceção de três características ($|\mathcal{V}_{ui}|$, $\mathbf{s}_{uv}$ e $\mathbf{p}_v$), nenhuma das outras piorou o resultado da previsão. Esses resultados foram mostrados na Tabela 4.3, que contém os resultados obtidos no estudo realizado sobre o impacto de todas as características na previsão da nota.

- *Qual o conjunto mínimo de evidências, entre as estudadas, mais eficaz para o problema dado?* Um possível conjunto mínimo de evidências contém as seguintes características: (i) Nota média do usuário alvo u ($\overline{r}_u$); (ii) Nota média do item ($\overline{r}_i$); (iii) Nota média da vizinhança ($\overline{r}_{\mathcal{V}_{ui}}$); (iv) Nota que vizinho v forneceu ao item i ($\mathbf{r}_{vi}$); e (v) Nota média do vizinho v ($\overline{\mathbf{r}}_v$). Usando apenas estas características, é possível obter um RMSE de 0,9281, um resultado quase tão bom quanto o melhor obtido neste trabalho.

5.2 Limitações

A principal limitação deste trabalho consiste na avaliação baseada em apenas uma coleção de dados. Para garantir que as conclusões observadas sejam válidas, mais experimentos devem ser realizados com diferentes coleções de referência. Outra limitação deste estudo é ser restrito apenas a métodos de filtragem colaborativa baseados em usuários. O mesmo estudo pode ser claramente adaptado para métodos baseados em itens e evidências relacionadas com fatores latentes. Finalmente, uma última limitação se deveu a um erro que ocorreu durante a transferência dos dados de *confiança entre usuários* de nossa implementação do Mahout para os nossos regressores. Como este erro foi detectado já na fase final de escrita da dissertação, prefirimos não usar estes dados como evidência adicional para a regressão.

5.3 Trabalhos Futuros

Dadas as limitações do atual trabalho, as primeiras linhas de estudo futuro devem se concentrar em eliminar tais limitações. Assim, em curto prazo, novos experimentos devem ser feitos considerando (a) a inclusão de evidências baseadas em confiança entre usuários para os regressores; (b) o uso de mais coleções, de preferência, coleções de referência e (c) outras classes de métodos de recomendação, em especial, filtragem colaborativa baseada em itens e métodos que exploram fatores latentes. Outro estudo interessante seria a extensão deste trabalho para métodos de regressão não linear como Support Vector Regression [Hall et al., 2009]. Outra direção seria a incorporação dos métodos estudados na infra-estrutura de ambientes escaláveis como o Mahout, de forma que versões distribuidas possam ser avaliadas.

Referências Bibliográficas

Aha, D. W.; Kibler, D. & Albert, M. K. (1991). Instance-based learning algorithms. Em *Machine Learning*, pp. 37--66.

Anderson, M.; Ball, M.; Boley, H.; Greene, S.; Howse, N.; Lemire, D. & e S. McGrath (2003). Racofi: A rule-applying collaborative filtering system. *IEEE/WIC COLA'03*.

Apache Software Foundation (2011). Apache mahout:: Scalable machine-learning and data-mining library.

Bell, R. & Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. Em *7th IEEE International Conference on Data Mining*, pp. 43--52.

Bouza, A.; Reif, G.; Bernstein, A. & Gal, H. (2008). Semtree: ontology-based decision tree algorithm for recommender systems. Em *Em International Semantic Web Conference*.

Breese, J.; Heckerman, D. & e Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 43--52.

Christakou, C. & Stafylopatis, A. (2005). A hybrid movie recommender system based on neural networks. *Proceedings of the 5th International Conference on Intelligent Systems Design an Applications*, pp. 500--505.

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. Em *IEEE Transactions on Information Theory*, pp. 21--27.

Cristianini, N. & Shawe-Taylor, J. (2000). An introduction to support vector machine and other kernel-based learning methods. Em *Cambridge University Press*, pp. 169--174.

Dean, J. & Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107--113.

Drucker, H.; Burges, C. J. C.; Kaufman, L.; Smola, A. & Vapnik, V. (1997). Support Vector Regression Machines. *Advances in neural information processing systems*, pp. 155--161.

Good, N.; Schafer, J. B.; Konstan, J. A.; Borchers, A.; Sarwar, B.; Herlocker, J. & Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. Em *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, AAAI '99/IAAI '99, pp. 439--446, Menlo Park, CA, USA. American Association for Artificial Intelligence.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P. & Witten, I. H. (2009). The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10--18.

Herlocker, J. L.; Konstan, J. A.; Terveen, L. G. & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5--53.

Liu, N. & Yang, Q. (2008). EigenRank: a ranking oriented approach to collaborative filtering. Em *Proceedings of the 31st Annual International ACM SIGIR Conference*.

Ma, H.; Lyu, M. R. & King, I. (2009). Learning to recommend with trust and distrust relationships. *Proceedings of the 3rd ACM Conference on Recommender Systems*, pp. 189--196.

Miyahara, K. & Pazzani, M. J. (2000). Collaborative filtering with the Bayesian classifier. Em *Pacific Rim International Conference on Artificial Intelligence*.

O'Donovan, J. & Smyth, B. (2005). Trust in recommender systems. *: Proceedings of the 10th International Conference on Intelligent User Interfaces*, pp. 945--962.

O'Donovan, J. & Smyth, B. (2006). Mining trust values from recommendation errors. Em *International Journal on Artificial Intelligence Tools*, pp. 945--962.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P. & Riedl, J. (1994). Grouplens: an open architecture for collaborative filtering of netnews. Em *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94, pp. 175--186, New York, NY, USA. ACM.

Ricci, F.; Rokach, L.; Shapira, B. & Kantor, P. (2011). *Recommender Systems Handbook*. Springer.

Rosenblatt, F. (1962). Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. *Washington DC: Spartan*.

Shi, Y.; Larson, M. & Hanjalic, A. (2010). List-wise Learning to rank with matrix factorization for collaborative filtering. Em *Proceedings of the 4th ACM Conference on Recommender Systems*, pp. 269--272.

Wang, Y. & Witten, I. H. (1997). Induction of Model Trees for Predicting Continuous Classes. *Proceedings of the Poster Papers of the European Conference on Machine Learning*, pp. 128--137.

Widrow, B. & Lehr, M. A. (1990). 30 years of adaptive neural networks - perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78:1415--1442 ST – 30 Years of Adaptive Neural Networ.

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80--83.

Xia, Z.; Dong, Y. & Xing, G. (2006). Support vector machines for collaborative filtering. *Proceedings of the 44th Annual Southeast Regional Conference*.

Ziegler, C. & Golbeck, J. (2007). Investigating correlations of trust and interest similarity. *Decision Support Systems*, pp. 460--475.

Zurada, j. (1992). Introduction to artificial neural systems. *West Publishing Co.*