



Universidade Federal de Pernambuco
Centro de Informática



Dissertação de Mestrado

Um Método de Aprendizagem Ativa em Sistemas de Filtragem Colaborativa

por

Ivan Romero Teixeira

(irt@cin.ufpe.br)

Orientador: Francisco de Assis Tenório de Carvalho

Co-orientador: Geber Lisboa Ramalho

Recife, 2002

Resumo

Quando se está diante de um problema de busca de informação em um assunto sobre qual não se tem o conhecimento apropriado, geralmente recorre-se à recomendação. No dia a dia estamos freqüentemente diante de situações como essa, onde procuramos obter informações para decidir a que restaurante ir, que filme assistir, ou que livro ler. Em qualquer uma dessas situações tem-se uma seleção a fazer dentro de um universo de escolhas sobre as quais busca-se mais informação sobre a qualidade ou relevância de cada possibilidade para que se possa tomar uma decisão, ou em outras palavras, busca-se uma informação mais direta, uma recomendação.

Os *Sistemas de Recomendação* são uma inovação recente que tem como objetivo ampliar o acesso e a qualidade das recomendações através da automatização do processo de recomendação. São sistemas de informação que coletam indicadores da preferência dos usuários e utilizam técnicas de *Filtragem de Informação* (FI) para fornecer uma visão personalizada da informação. Uma técnica de FI comum nesses sistemas é a *Filtragem Colaborativa*, que usa similaridade entre os perfis dos usuários para recomendar a um usuário a informação que outros usuários de preferência similar a ele avaliaram positivamente. Uma importante questão relativa aos sistemas de recomendação está em como obter as preferências dos usuários. A forma mais comum de obter a opinião do usuário é através de avaliações explícitas sobre a relevância de uma informação indicada por uma nota. Entretanto esse processo pode se tornar tedioso para um usuário uma vez que o número de avaliações necessárias para aprender o suficiente sobre os interesses do mesmo pode ser muito grande.

Neste trabalho sugerimos um processo de aquisição do perfil dos usuários através da *aprendizagem ativa*. A aprendizagem ativa é o paradigma no qual o algoritmo de aprendizagem tem algum controle sobre os exemplos usados no seu treinamento, permitindo a escolha dos exemplos mais informativos para seu aprendizado. Mais precisamente, neste trabalho abordamos o problema de realizar aprendizagem ativa no contexto de um sistema de recomendação com filtragem colaborativa, de forma a (1) minimizar a quantidade de avaliações do usuário requeridas para atingir uma qualidade de recomendação desejada, ou (2) maximizar a qualidade das recomendações para uma quantidade fixa de avaliações. Para isso criamos um novo método *ActiveCP* baseado nos critério de controvérsia e popularidade (CP) dos itens que permite que o algoritmo de filtragem colaborativa selecione os itens que quando avaliados resultam num maior ganho de informação sobre o perfil do usuário. O método usado para selecionar os itens que os usuários devem avaliar foi capaz de melhorar o aprendizado sobre a preferência do usuário quando comparado a uma seleção aleatória dos itens avaliados, fornecendo melhores recomendações com um número bem menor de avaliações. A aplicação desse método é apresentada em um sistema de recomendação desenvolvido.

Palavras Chaves: sistemas de recomendação, filtragem de informação, filtragem colaborativa, aprendizagem ativa

Abstract

When one faces a problem of information search in a subject for which one doesn't have appropriate knowledge about, usually one appeals to recommendation. In every day we are usually ahead of situations like this, where we search for information to decide which restaurant to go, which movie to watch, or which book to read. In any of those situations there is a selection to make in a universe of choices for which one searches more information about the quality or relevance of each possibility, so that one can make a decision, or in other words, one searches a more direct kind of information, a recommendation.

Recommender Systems are a recent innovation that has the objective of enlarging the access and quality of recommendations by automating the process of recommendation. They are information systems that collect indicators of users preferences and use *Information Filtering* (IF) techniques to provide a personalized view of the information. A common technique of IF in those systems is the *Collaborative Filtering*, which uses the similarities among users profiles to recommend to a user the information that other users with preferences similar to him evaluated positively. One important topic in recommender systems is how to obtain users preferences. The most common way to obtain users opinion is through explicit evaluations about the relevance of an information indicated by a grade. However this process can become tedious for the user once the number of evaluations required to learn enough about his interests can be very large.

In this work we suggest a process of acquisition of users profile through *active learning*. Active learning is the paradigm where the learning algorithm has some control over the examples used on its training. More precisely, in this work we deal with the problems of performing active learning in the context of a recommender system with collaborative filtering, so as (1) to minimize the quantity of user evaluations required to reach a quality target of recommendation, or (2) to maximize the quality of recommendations for a fixed quantity of user evaluations. For this we create a new method *ActiveCP* based on the criterions of controversy and popularity (CP) of items that enables the collaborative filtering algorithm to select the items that when evaluated result in a greater information gain about the user profile. The method used to select the items users should evaluate was capable to improve the learning about user preferences when compared to a random selection of evaluated items, providing better recommendations with a much lower number of evaluations. The application of this method is demonstrated in a recommender system developed.

Keywords: recommender systems, information filtering, collaborative filtering, active learning.

Índice

CAPÍTULO 1	1
<i>Introdução</i>	
1.1 MOTIVAÇÃO	2
1.2 OBJETIVOS	3
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	4
CAPÍTULO 2	8
<i>Metodologias de busca de informação</i>	
2.1 RECUPERAÇÃO DE INFORMAÇÃO	9
2.2 FILTRAGEM DE INFORMAÇÃO	10
2.3 SISTEMAS DE RECOMENDAÇÃO	12
2.4 CONCLUSÃO	15
CAPÍTULO 3	16
<i>Tecnologias de Sistemas de Recomendação</i>	
3.1 SISTEMAS DE RECOMENDAÇÃO NÃO PERSONALIZADOS	17
3.2 SISTEMAS DE RECOMENDAÇÃO PERSONALIZADOS	19
3.2.1 Técnicas de Filtragem de Informação.....	19
3.2.2 Filtragem Baseada em Conteúdo	21
3.2.3 Filtragem Colaborativa.....	23
3.2.4 Filtragem Híbrida	24
3.3 MEDINDO A QUALIDADE DAS RECOMENDAÇÕES	26
3.3.1 Tarefas de um Sistema de Recomendação	26
3.3.2 Métricas	29
3.4 CONCLUSÃO	34

CAPÍTULO 436

Coletando as preferências do usuário

4.1	INDICADORES IMPLÍCITOS.....	37
4.2	INDICADORES EXPLÍCITOS.....	39
4.3	INDICADORES IMPLÍCITOS VERSUS INDICADORES EXPLÍCITOS.....	40
4.4	APRENDIZAGEM ATIVA.....	41
4.5	APRENDIZAGEM ATIVA EM SISTEMAS DE FI.....	42
4.6	CONCLUSÃO.....	45

CAPÍTULO 546

Filtragem Colaborativa

5.1	DEFINIÇÃO DO PROBLEMA	47
5.2	ALGORITMOS DE FILTRAGEM COLABORATIVA.....	48
5.3	FILTRAGEM COLABORATIVA COM VIZINHOS MAIS PRÓXIMOS	51
5.4	ALGORITMO DE FILTRAGEM COLABORATIVA	53
5.4.1	<i>Similaridade entre os usuários.....</i>	53
5.4.2	<i>Seleção dos usuários</i>	55
5.4.3	<i>Geração de previsões.....</i>	55
5.4.4	<i>Algoritmo de Predição</i>	56
5.5	CONCLUSÃO.....	59

CAPÍTULO 660

ActiveCP: Um método de aprendizagem ativa para filtragem colaborativa

6.1	<i>SELECTIVE SAMPLING PARA FC COM VIZINHOS MAIS PRÓXIMOS</i>	61
6.1.1	<i>Seleção de exemplos em KNN</i>	61
6.1.2	<i>Seleção de exemplos em FC com vizinhos mais próximos.....</i>	64
6.1.3	<i>Critérios de seleção de itens em FC com vizinhos mais próximos</i>	65

6.2	CONTROVÉRSIA.....	66
6.2.1	<i>O que é controvérsia ?</i>	69
6.2.2	<i>Como medir a controvérsia ?</i>	71
6.2.3	<i>Controvérsia Global versus Controvérsia Local</i>	73
6.2.4	<i>Cobertura versus Intensidade da Controvérsia</i>	74
6.3	POPULARIDADE	75
6.4	ANÁLISE EMPÍRICA	76
6.4.1	<i>Base de Dados</i>	77
6.4.2	<i>Métricas</i>	77
6.4.3	<i>Organização dos experimentos</i>	78
6.4.4	<i>Teste de hipótese.....</i>	80
6.4.5	<i>Performance</i>	81
6.4.6	<i>ActiveCP</i>	89
6.4.7	<i>Análise do método ActiveCP</i>	95
6.5	CONCLUSÃO	100
CAPÍTULO 7		102
<i>GEP: Um sistema de recomendação para entretenimento</i>		
7.1	VISÃO GERAL DO SISTEMA	103
7.2	AMBIENTE DE DESENVOLVIMENTO	104
7.3	PROJETO E IMPLEMENTAÇÃO	106
7.4	ARQUITETURA DO SISTEMA	107
7.4.1	<i>Camada de Comunicação</i>	108
7.4.2	<i>Camada de Negócio</i>	108
7.4.3	<i>Camada de Dados</i>	110
7.5	APRESENTAÇÃO DO GEP	111
7.6	CONCLUSÃO	118

CAPÍTULO 8 120***Conclusões***

8.1 CONTRIBUIÇÕES..... 121

8.2 TRABALHOS FUTUROS 123

Lista de Figuras

Figura 2.1 - Modelo do processo de recomendação	13
Figura 3.1 – Exemplo da descrição de um filme	21
Figura 3.2 – Problema de filtragem baseada em conteúdo	22
Figura 3.3 – Curvas da distribuição dos valores para elementos relevantes e não relevantes	30
Figura 3.4 – Exemplo de uma curva ROC.....	31
Figura 3.5 – Exemplos de curvas ROC da cobertura vs. ruído	31
Figura 4.1 – Tela do sistema MovieLens.	39
Figura 4.2 – Modelo de avaliação de uma sistema de filtragem passivo.....	43
Figura 4.3 – Modelo de avaliação de um sistema de filtragem ativo.....	44
Figura 5.1 – Exemplo da matriz de avaliações de usuários sobre itens em um intervalo de 1 a 5.....	47
Figura 5.2 – Vizinhança utilizada na predição dos itens A e B para o usuário alvo.....	52
Figura 5.3 – Algoritmo de filtragem colaborativa com vizinhos mais próximos.	57
Figura 6.1 – Diagrama de Voronoi para um conjunto de pontos.....	63
Figura 6.2– A relação do usuário alvo com os demais usuário antes e depois da avaliação do item i	68
Figura 6.3 - A relação do usuário alvo com os demais usuário antes e depois da avaliação do item j	69
Figura 6.4 - Exemplos de distribuições das avaliações para itens com escores entre 1 a 5	70
Figura 6.5 – Variância para as distribuições mostradas na figura 6.4	71
Figura 6.6 – Algoritmo do teste de seleção dos itens que um usuário avalia	79
Figura 6.7 – Metodologia de teste dos algoritmos de seleção	80
Figura 6.8 – Gráfico da performance das predições usando as avaliações dos itens selecionados pelo critério da controvérsia calculada em amostras de 100 e 1000 usuário e em todos os usuários.	83
Figura 6.9 – Gráficos da performance das predições usando as avaliações dos itens selecionados com a controvérsia medida em uma amostra de 1000 usuário e aleatoriamente.....	84
Figura 6.10 – Gráficos da performance das predições usando as avaliações dos itens selecionados com a controvérsia medida em uma amostra de 1000 usuários, com a popularidade total e aleatoriamente....	86
Figura 6.11 – Gráficos comparando as metodologia de seleção com o critério CP e aleatória.	94
Figura 6.12 – Gráfico comparando as metodologias de seleção baseadas na controversia, popularidade e na combinação das duas através de CP 5.	96
Figura 6.13 – Gráficos com o número de avaliações necessárias para obter a performance desejada utilizando a seleção aleatória das avaliações e a seleção pelo critério CP 5.	97
Figura 6.14 – Razão N_a/N_{CP5} entre o número de itens selecionados aleatoriamente N_a e selecionados pelo critério CP 5 para obter uma performance desejada nas métricas ROC e Breese	99
Figura 6.15 – Quantidade de avaliações que devem ser selecionadas aleatoriamente para obter performance média de predições igual à obtida pela quantidade de avaliações selecionadas pelo critério CP 5.....	99
Figura 7.1- Ontologia de eventos de entretenimento.....	104

Figura 7.2 – Modelo geral do sistema GEP	106
Figura 7.3 – Arquitetura do sistema GEP	107
Figura 7.4 – Interface do Módulo de Personalização	109
Figura 7.5 – Interface do Módulo de Aquisição de Perfil	110
Figura 7.6 – Tela inicial do sistema GEP	112
Figura 7.7 – Consulta à programação do canal HBO.	113
Figura 7.8 – Tela inicial para o usuário conectado	114
Figura 7.9 – Recomendação de programas para um usuário.	115
Figura 7.11 – Tela de preferências de programa.	117
Figura 7.12 – Tela de aprendizagem ativa do perfil do usuário.....	118

Lista de Tabelas

Tabela 6.1 – Tabela de ROC e Breese para controvérsia 100 e 1000 usuários e total.....	83
Tabela 6.2 – ROC: controvérsia 1000 vs. aleatória	85
Tabela 6.3 – Breese: controvérsia 1000 vs. aleatória	85
Tabela 6.4 – Tabela de ROC e Breese para controvérsia 1000, popularidade total e aleatória.	87
Tabela 6.5 – ROC: popularidade total vs. aleatório.....	88
Tabela 6.6 – Breese: popularidade total vs. aleatório	88
Tabela 6.7 – Exemplos da distribuição e variância das avaliações de filmes da base de dados	90
Tabela 6.8 – Tabela dos pesos utilizados no critério CP de seleção das avaliações.	92
Tabela 6.9 – Performance ROC de CP0 a CP10	92
Tabela 6.10 – Performance Breese de CP0 a CP 10.....	93
Tabela 6.11 – ROC: CP5 vs. aleatório.....	94
Tabela 6.12 – Breese: CP5 vs. aleatório.....	95
Tabela 6.13 – CP5 vs. perfil completo	98
Tabela 7.1 – Ferramentas utilizadas no desenvolvimento do sistema GEP	105

Capítulo 1

Introdução

1.1 Motivação

A evolução recente dos meios de comunicação permitiu o início de uma nova era da informação. O acesso à informação permitiu grandes mudanças que beneficiaram a vida das pessoas.

O acesso à informação entretanto não veio sem uma contrapartida. O excesso de informação passou a ser um problema quando se necessita encontrar a informação apropriada. Uma pessoa freqüentemente se encontra diante de inúmeros livros, filmes, artigos, páginas *Web*, programas de televisão, notícias e muitos outros tipos de informação dos quais apenas uma pequena parcela lhe interessa e portanto se vê obrigada a buscar dentro de uma quantidade muito grande de opções aquelas apropriadas às suas necessidades. Esse quadro torna frustrante a tarefa de encontrar uma informação, uma vez que o custo de obter uma informação pode torna-se maior do que o seu benefício.

Diante desse problema surge a proposta de automatizar a busca por informação através de sistemas de informação capazes de auxiliar as pessoas na busca pela informação apropriada. Os sistemas de informação atualmente utilizam metodologias de recuperação de informação (RI) e filtragem de informação (FI) para encontrar a informação requisitada pelos usuários. FI é uma metodologia que utiliza o conhecimento sobre o perfil de um usuário para decidir sobre a relevância de uma informação para este.

Atualmente um tipo de sistema de informação capaz de adquirir o perfil de um usuário e recomendar a informação apropriada à seus gostos e necessidades, vem se tornando cada vez mais popular. Os chamados *sistemas de recomendação* utilizam filtragem de informação para automatizar o processo de fornecer recomendações. Dentre as filtrações usadas para implementar tais sistemas a *filtragem colaborativa* (FC) tem apresentado grande aceitação. Essa técnica de filtragem identifica as relações entre os perfis dos usuários para realizar a filtragem automatizando o processo pelo qual pessoas com necessidades e preferências similares auxiliam umas as outras na busca pela informação.

Uma das principais tarefas para um sistema de recomendação é aprender sobre as necessidades e gostos de um usuário. Para isso, o sistema adquire indicadores do interesse do usuário pela informação. Esses indicadores são frequentemente obtidos através de avaliações explícitas dos itens pelos usuários. Em um sistema de recomendação de filmes, por exemplo, o interesse de um usuário pode ser indicado por uma nota que ele atribui ao filme. Entretanto, um sistema que requer indicadores de preferências explícita de um usuário exige do mesmo a tarefa de realizar uma quantidade substancial de avaliações antes que o sistema possa aprender sobre seu perfil. A necessidade de fornecer um número cada vez maior de avaliações para obter um pequeno ganho na qualidade das recomendações pode afastar os usuários do sistema.

Dá-se o nome de aprendizagem ativa em um algoritmo de aprendizagem supervisionada ao processo pelo qual o algoritmo tem algum controle sobre quais exemplos serão usados no seu treinamento. Essa metodologia é usada em problemas nos quais a classificação de novos exemplos é dispendiosa e requer-se que seja minimizado o número de avaliações necessárias para generalizar o conceito que se quer aprender. A tarefa de aprender sobre o perfil de um usuário em um sistema de recomendação é uma tarefa que pode beneficiar-se da aprendizagem ativa, pois os exemplos usados para o aprendizado desse perfil exige o trabalho manual do usuário ao avaliar um item.

1.2 Objetivos

Neste trabalho sugerimos uma metodologia de aprendizagem ativa do perfil dos usuários através da seleção dos itens para os quais é requisitada a sua avaliação, com o objetivo de minimizar a quantidade de avaliações que um usuário deve fornecer para que o sistema possa conhecer o seu perfil de forma suficiente para fornecer-lhe boas recomendações. Em resumo, os principais objetivos desta dissertação são os seguintes:

1. Desenvolver uma nova metodologia de aprendizagem ativa do perfil dos usuários em um sistema de filtragem colaborativa.
2. Comprovar a eficiência do método de seleção, comparando a precisão das recomendações utilizando o perfil obtido com o método desenvolvido com a precisão das recomendações utilizando o método convencional de obtenção do perfil.
3. Desenvolver um sistema de recomendação para mostrar o interesse e a importância da aprendizagem ativa em um sistema de recomendação.

Analizamos o algoritmo de filtragem colaborativa com vizinhos mais próximos, um dos algoritmos de filtragem de maior aceitação e aplicação em sistemas de recomendação, e desenvolvemos um método original de selecionar os itens que um usuário avalia, com o objetivo de aprender rapidamente sobre o perfil do usuário sem exigir dele uma quantidade grande de avaliações e conseqüentemente melhorando a qualidade das recomendações em um sistema de recomendação.

Para desenvolver um novo método de aprendizagem ativa do perfil do usuário, introduzimos critérios de *controvérsia* e de *popularidade* de um item que foram usados pelo sistema. Esses dois critérios foram combinados no desenvolvimento do *ActiveCP*, um método de aprendizagem ativa do perfil dos usuários. Esse método foi aplicado com sucesso na seleção das avaliações dos usuários de um sistema recomendação de filmes.

Por último apresentamos o *GEP* (Guia de Entretenimento Personalizado), um sistema de recomendação que utiliza filtragem colaborativa e aprendizagem ativa de perfil para aprender sobre o perfil de um usuário e recomendar a ele os programas de entretenimento de acordo com a sua preferência.

1.3 Organização da dissertação

O restante dessa dissertação está organizado em 7 capítulos. O conteúdo dos capítulos é apresentado a seguir.

O **Capítulo 2** apresenta os métodos de busca de informação automatizados usados atualmente. São apresentadas as diferenças entre filtragem de informação e recuperação de informação e em que situações a filtragem de informação é mais apropriada. Por último é apresentada a recomendação como meio de obter informação relevante e os sistemas de recomendação que automatizam o processo de recomendação.

O **Capítulo 3** fala do estado da arte em sistemas de recomendação e na diferença entre sistemas personalizados e não personalizados. A tecnologia de filtragem de informação usada nos sistemas de recomendação personalizada são apresentadas. Finalmente é feito uma análise sobre como avaliar um sistema de recomendação e uma descrição das métricas de avaliações usadas nesse trabalho.

O **Capítulo 4** descreve as formas que a preferência dos usuários pode ser obtida e quais as vantagens e desvantagens de cada uma delas. Esse capítulo também descreve como a aprendizagem ativa pode ser utilizada em sistemas de recomendação para obter o perfil dos usuários de forma mais eficiente.

O **Capítulo 5** descreve o estado da arte em filtragem colaborativa. O algoritmo de filtragem colaborativa com vizinhos mais próximos utilizado na dissertação é descrito e a sua ordem de complexidade avaliada.

O **Capítulo 6** descreve a metodologia de aprendizagem ativa para filtragem colaborativa desenvolvida. É apresentada a noção de ganho de informação da avaliação de um item e a controvérsia e popularidade dos itens são sugeridos como critérios de seleção. O método *ActiveCP* de seleção dos itens que o usuário avalia é apresentado e são feitos testes para medir a performance dos métodos de seleção criados.

O **Capítulo 7** descreve o sistema de recomendação desenvolvido, o GEP. O sistema apresentado utiliza a aprendizagem ativa desenvolvida para aprender o perfil do usuário e a filtragem colaborativa para gerar recomendações. O sistema é mostrado através de telas de sua utilização.

O **Capítulo 8** resume as conclusões do trabalho, apresentando as principais contribuições e os possíveis trabalhos futuros.

Capítulo 2

Metodologias de busca de informação

O problema de sobrecarga de informação não é recente. Antes mesmo do surgimento de tecnologias de grande capacidade de disseminar informação como, por exemplo, a Internet, já era necessário uma busca automatizada da informação correta para cada necessidade de informação. A busca por informação tornou-se ainda mais importante nos últimos anos com o crescimento das tecnologias automáticas de armazenamento, produção e disseminação de informação.

Existem diferentes tecnologias que são usadas para tratar o problema de sobrecarga de informação. Essas tecnologias podem ser divididas em duas categorias: *recuperação de informação* e *filtragem de informação* [Belkin 92].

2.1 Recuperação de Informação

Muitos dos problemas de sobrecarga de informação vem sendo resolvidos de forma satisfatória por técnicas de *recuperação de informação* (RI). A RI baseia-se em uma consulta formulada pelo próprio usuário sobre o tópico de seu interesse. Este processo tem hoje técnicas avançadas para realizar busca de informações não estruturadas em documentos texto e HTML [Baeza-Yates 99].

A RI em dados não estruturados geralmente utiliza técnicas como indexação dos documentos de acordo com a parte relevante de seu conteúdo, classificação dos resultados usando uma ampla variedade de heurísticas e avaliação da relevância das palavras que aparecem no documento de acordo com sua frequência e identificação de sinônimos. Exemplos de sucesso de sistemas de RI para dados não estruturados são os engenhos de busca na Internet como o Google (www.google.com) e o Yahoo (www.yahoo.com).

Para dados estruturados, a RI indexa os itens de acordo com os atributos considerados relevantes e realiza pesquisa tendo por base esses atributos. Uma base de filmes, por exemplo, pode ser indexada por atores, diretores, gênero e outros atributos que podem ser utilizados pelos usuários para realizar busca na base de filmes. Um exemplo de sistema de RI em uma base de dados estruturados é o

Internet Movie Data Base (www.imdb.com) que possui uma base de dados sobre filmes que pode ser pesquisada por atributos como nome, diretor ou gênero.

A RI é bastante útil para o usuário que sabe descrever o que está buscando de forma compatível com a descrição do conteúdo que foi criada durante a indexação. Esta técnica tem como característica a busca de informação através da associação dos itens com a consulta do usuário. Em nenhum momento os sistemas de RI distinguem seus usuários e suas diferentes características no momento de realizar uma busca. Para alguns casos a necessidade de informação é bem atendida por um sistema de RI, entretanto, em muitos outros é necessária a identificação das características e preferências individuais de cada usuário para realizar uma busca.

2.2 Filtragem de Informação

Para os problemas que exigem uma solução mais dirigida às características do usuário é interessante construir um sistema capaz de realizar *filtragem de informação* (FI) [Herlocker 00]. A FI diferencia-se de RI pela identificação das necessidades e gostos persistentes dos usuários, e utiliza essas características na busca de informação para o usuário. Em um sistema de FI, os usuários possuem objetivos e necessidades que são relativamente estáveis e persistentes. O usuário passa a ser caracterizado por esses objetivos e necessidades, um interesse regular de informação que pode variar lentamente com o tempo. Como exemplo de situações onde é possível beneficia-se de um sistema de FI, pode-se citar uma pessoa executando uma tarefa sistemática, de longa duração e de necessidade contínua de informação pertinente como, por exemplo, um pesquisador ou um funcionário trabalhando em determinado assunto. Uma pessoa buscando informação sobre entretenimento possui características semelhantes quanto à busca por informações sobre assuntos como livros, filmes ou música, assuntos sobre os quais as pessoas possuem um interesse relativamente estável.

Para relacionar a informação apropriada a cada usuário, um sistema de FI possui uma representação do perfil deste usuário que reflete sua necessidade de

conhecimento. O sistema tem como tarefa buscar a informação em uma base de informação e apresentar ao usuário a parte desta mais pertinente ao seu perfil, ou em alguns casos atuar em um fluxo contínuo de informação, como, por exemplo, uma caixa de entrada de e-mail, e filtrar o que não é do seu interesse. Essas informações são usadas e/ou avaliadas com relação a quanto elas atendem às suas necessidades de informação. As avaliações por sua vez podem levar a uma modificação da representação do perfil do usuário na tentativa de melhor adequar o sistema aos interesses dele.

Algumas diferenças entre RI e FI [Belkin 92] são:

- Enquanto RI preocupa-se com a utilização única do sistema, por uma pessoa com um objetivo momentâneo representado em uma consulta, FI preocupa-se com utilizações frequentes do sistema por pessoas com interesses e necessidades de longo termo.
- Enquanto RI tem como especificação da informação requerida uma consulta formulada pelo usuário, FI assume o perfil do usuário como especificação da informação requerida.
- Enquanto RI preocupa-se em colecionar e organizar a informação para retorná-la a uma consulta adequada, FI preocupa-se em coletar as preferências do usuário e entregar a ele a informação adequada.
- Enquanto em RI preocupa-se com episódios independentes de busca de informação, FI preocupa-se com mudanças na especificação da informação requerida ao longo de série de eventos de busca.

FI e RI portanto procuram resolver o mesmo problema: a necessidade de encontrar informação apropriada dentro de uma grande massa de informação, mas com abordagens diferentes essencialmente na forma com que essa necessidade é representada e aplicada nesta busca.

2.3 Sistemas de Recomendação

Quando uma pessoa está diante de um problema de busca de informação em um assunto sobre o qual não tem o conhecimento apropriado, ela geralmente recorre às recomendações. No dia a dia estamos freqüentemente diante de situações como essa, onde procuramos obter informações para decidir a que restaurante ir, que filme assistir, ou que livro ou notícia ler. Para qualquer desses casos, tem-se uma seleção a fazer dentro de um universo de escolhas nas quais busca-se mais informação sobre a qualidade ou relevância de cada possibilidade, para que então se possa tomar uma decisão. Em outras palavras, busca-se uma forma bem mais direta de informação, uma recomendação. As pessoas buscam recomendações como uma solução mais fácil porque geralmente estão diante de problemas onde o universo de escolhas é muito grande, às vezes em uma extensão em que uma pessoa apenas não seja capaz de conhecer e avaliar todas as possibilidades existentes para escolher a melhor ou pelo menos uma satisfatória.

Fornecer e receber recomendações é um ato de comunicação comum que já existe há muito tempo. Geralmente encontram-se recomendações sobre muitos tópicos conversando com amigos ou colegas, lendo artigos de especialistas ou simplesmente observando o comportamento das pessoas (como comprar o livro mais vendido baseando-se na sua popularidade). Os *sistemas de recomendação* são uma inovação recente que tem como objetivo ampliar o acesso e a qualidade das recomendações através da automatização do processo pelo qual as recomendações são formadas e entregues às pessoas apropriadas. Os sistemas de recomendação são sistemas de informação que usualmente coletam indicadores sobre a preferência dos usuários e utilizam técnicas de FI para fornecer uma visão personalizada da sua base de informação, i. e. a informação sobre opções nas quais o usuário tem maior chance de estar interessado, dado a informação coletada sobre sua preferência.

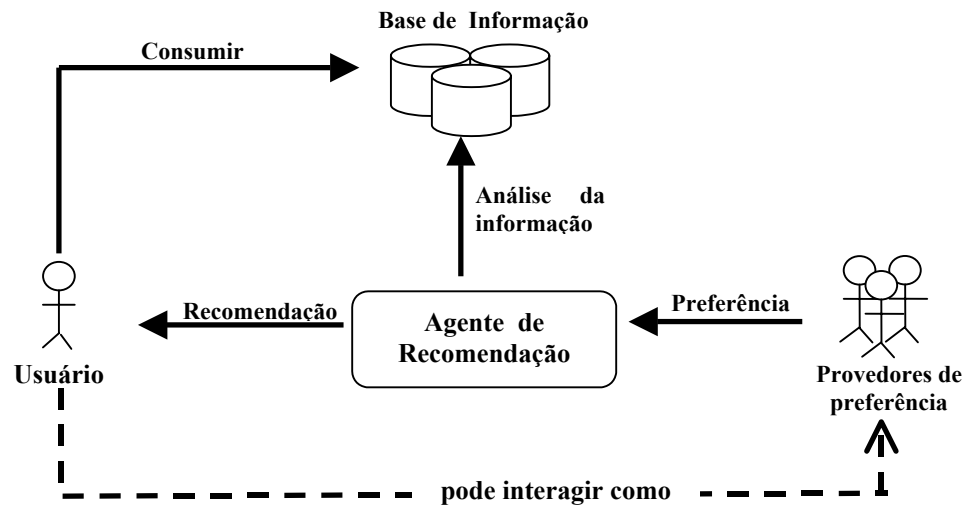


Figura 2.1 - Modelo do processo de recomendação

A figura 2.1 apresenta um modelo geral de um processo de recomendação. O modelo apresentado é análogo ao apresentado por Terveen et al. [Terveen 01] com algumas adaptações. Os componentes do processo são:

- **Usuário** : o objetivo de todo processo de recomendação é fornecer auxílio para o usuário no processo de busca por informação. Ele pode interagir passivamente recebendo apenas recomendações ou mais ativamente, assumindo o papel de provedor de preferência.
- **Base de Informação**: é onde está contido o universo de alternativas onde o usuário deseja encontrar a informação relevante. Essa base de informações pode estar em um repositório centralizado ou distribuído, como no caso da Internet ou mesmo no ambiente real que o cerca.
- **Agente de Recomendação**: aquele que fornece a recomendação assume o papel do Agente de Recomendação. A recomendação pode ser feita manualmente por um especialista humano. Os sistemas de recomendação buscam assumir o papel do Agente de Recomendação, automatizando este processo.
- **Provedores de Preferência**: o processo de recomendação baseia-se em indicadores humanos de preferência. Os provedores podem ser em grande

número formando uma comunidade na qual o Agente de Recomendação pode se basear para gerar recomendações ou pode ser o próprio usuário.

A recomendação funciona como uma visão ou filtro da informação contida na base de informação. Uma recomendação pode ser baseada na opinião do agente de recomendação, na preferência do usuário ou mesmo na opinião de uma comunidade [Claypool 00]. Pode ser voltada para a comunidade a que o usuário pertence ou para cada um individualmente. Finalmente, a forma com que a recomendação é apresentada pode variar desde um complexo texto, como em um artigo de um especialista, até uma nota ou uma simples lista de recomendações [Terveen 01]. Com auxílio destas recomendações o usuário se volta para a base de informação para selecionar o que lhe convém.

O agente de recomendação tem acesso à base de informação onde é capaz de analisar as opções no intuito de produzir as recomendações. A análise da informação tem o intuito de extrair na base a parte relevante para o usuário ou comunidade alvo.

As atividades de recomendação que interessam neste trabalho são as automatizadas através de um sistema de recomendação. Esses sistemas buscam automatizar processos de recomendação, muitas vezes análogos a modelos de recomendação já existentes sem o auxílio desses sistemas.

A preferência é o indicador do perfil do usuário que busca recomendação. No processo de recomendação, a forma com que a preferência é coletada pode variar bastante. O usuário pode ser requisitado a fornecer sua preferência ou o sistema pode tomar a iniciativa de buscar a preferência do usuário ou da comunidade através de observações sobre eles. Neste trabalho propomos um método novo capaz de aprender rapidamente sobre o perfil do usuário. O nosso método será apresentado no capítulo 7. Antes disso falaremos das tecnologias de sistemas de recomendação e filtragem de informação que serão pertinentes ao nosso problema.

2.4 Conclusão

Esse capítulo teve como principal objetivo contextualizar as abordagens atuais para problemas de busca de informação dentro de uma massa de dados muito grande. Foram apresentadas as tecnologias de busca de informação existentes. A recuperação de informação e filtragem de informação foram introduzidas, e discutimos em que situações a filtragem de informação é mais apropriada. A recomendação é apresentada como uma forma mais eficiente e direta de fornecer a informação necessária a pessoa. Mostramos um modelo do processo pelo qual as recomendações são geradas e entregues às pessoas. Os sistemas de recomendação são apresentados como uma solução para automatizar o processo de recomendação, tornando o processo mais eficiente e acessível a um maior número de pessoas.

Capítulo 3

Tecnologias de Sistemas de Recomendação

Os sistemas de recomendação podem ser divididos em duas categorias de acordo com a forma com que as recomendações são produzidas. A primeira categoria é formada pelos sistemas não personalizados, que buscam as recomendações de terceiros e as entrega para a pessoa ou comunidade apropriada. A segunda categoria é formada pelos sistemas de recomendação personalizados, que aprendem sobre a preferência do usuário e geram recomendações adaptadas ao seu perfil. A seguir são apresentadas características e exemplos de sistemas de recomendação divididos pela forma com que as recomendações são produzidas.

3.1 Sistemas de Recomendação Não Personalizados

São os sistemas de recomendação que tem como objetivo mediar o processo de recomendação. Esses sistemas têm como finalidade dar suporte às recomendações, fazendo apenas o papel de coleta de preferência e/ou distribuição de recomendações.

O sistema Tapestry [Goldberg 92] é frequentemente referido como o primeiro sistema de recomendação de que se tem conhecimento. O sistema Tapestry tem como objetivo prover suporte para recomendações de mensagens eletrônicas de grupos de notícia ou correspondência eletrônica. Ele permite que os usuários avaliem uma mensagem como “bom” ou “ruim” ou que postem alguma anotação sobre a mensagem. As mensagens são armazenadas em um repositório junto com as avaliações e anotações, e os usuários podem recuperá-las de acordo com o seu conteúdo ou com a opinião dos demais usuários. Por exemplo, um usuário poderia fazer uma consulta do tipo “mostre-me as mensagens que Ivan Teixeira avaliou como bom” ou ainda “mostre-me as mensagens que os membros de meu grupo responderam”. O Tapestry é útil para mediar o processo de recomendação, mas o sistema não identifica a necessidade de informação do usuário. O próprio usuário deve ser capaz de buscar recomendações baseando-se na semelhança de necessidades de informações e preferências que ele acredita ter com um ou mais usuários do sistema.

Na Internet a noção de recomendação é comum em páginas especializadas ou pessoais onde é freqüente encontrar recomendações de páginas sobre assuntos específicos. O About.com massificou esse tipo de recomendação especializada e impessoal [Terveen 01]. Este sistema é baseado no serviço de guias, que são usuários interessados em fornecer recomendações de páginas sobre um tópico do qual tem conhecimento. Os guias são selecionados de acordo com sua capacidade de achar informação interessante para os demais usuários. O About.com funciona como um sistema capaz de intermediar e massificar o processo de geração de recomendações por pessoas especializadas e a distribuição destas através da Internet.

É comum que as pessoas busquem recomendações através da observação da preferência das pessoas sobre o assunto que tem interesse. Quando uma pessoa acompanha a lista de mais vendidos de uma livraria ou vai a um restaurante que está sempre cheio, ela está observando a avaliação implícita de qualidade e relevância da comunidade interessada no assunto. Alguns sistemas buscam exercer um processo análogo, analisando registros digitais como documentos, páginas na Internet ou grupos de notícias em busca de indicadores de preferência para gerar recomendações, num processo referido como Mineração de Dados Sociais [Terveen 01]. O sistema PHOAKS [Terveen 97] analisa mensagens em grupos de notícia, busca referências para páginas na Web e identifica recomendações dos usuários dentre as referências. As recomendações dos usuários são combinadas para construir uma lista de recomendações de páginas relevantes e populares para cada grupo de notícias. Alguns sistemas de RI fazem esse tipo de mineração de dados sociais nas páginas pesquisadas na *Web*. O algoritmo *PageRank* [Page 98], usado no popular sistema de RI Google (www.google.com), considera dados como o número de referências existentes para uma página ou o número de vezes que uma página é selecionada em resultados de buscas como indicadores de preferência para agrupar e ordenar os resultados de uma busca.

Ainda que sigam um processo de recomendação, os sistemas de recomendação não personalizados apresentam como característica principal a incapacidade de

aprender sobre as preferências e necessidades de um usuário e de gerar automaticamente recomendação associadas ao seu perfil. Os sistemas de recomendação personalizados buscam atingir esse grau de personalização utilizando-se de técnicas de FI e de aprendizagem de máquina.

3.2 Sistemas de Recomendação Personalizados

Os sistemas de recomendação capazes de identificar e aprender as preferências e necessidades de um usuário, gerando recomendações apropriadas ao seu perfil são chamados sistemas de recomendação personalizados. Estes sistemas identificam cada usuário de forma individual e coletam suas preferências que são armazenadas em uma representação interna. O sistema utiliza técnicas de aprendizagem de máquina e FI para, a partir do perfil do usuário, gerar recomendações apropriadas aos interesses deste. A seguir discute-se as tecnologias utilizadas na construção de sistemas de recomendação personalizados.

3.2.1 Técnicas de Filtragem de Informação

A filtragem de informação é uma tarefa comum em ambientes onde o fluxo de informação é grande. Malone et al. [Malone 87] estudaram como as pessoas lidam com grande quantidade de informação eletrônica e identificaram três formas de filtragem que poderiam ser utilizadas num sistema de filtragem automática de mensagens eletrônicas. A classificação feita para as técnicas de FI [Malone 87], ainda que idealizada voltada para o problema de filtragem de mensagens eletrônicas, é extensível para filtragem de informações em geral e é freqüentemente usada para classificar os algoritmos de FI aplicados a variados tipos de informação. As formas de executar FI são classificadas em: *Filtragem Baseada em Conteúdo*, *Filtragem Econômica* e *Filtragem Colaborativa*.

Filtragem Baseada em Conteúdo ou *Filtragem Cognitiva* filtra informações baseadas no conteúdo dos elementos filtrados. Por exemplo, um filtro de e-mail que apaga automaticamente e-mails com palavras indesejadas, ou que envia um sinal

quando chega um e-mail com palavras-chaves contidas em um dicionário de assuntos importantes para o usuário, são exemplos de filtragem cognitiva. Por ter sua origem em pesquisas de RI é comum aplicar técnicas como indexação e análise de conteúdo nesse tipo de filtragem.

Filtragem Econômica filtra informação baseada na avaliação do custo para consumir e/ou produzir o item e o benefício do uso deste. Na filtragem econômica existe a necessidade de estimar o custo de consumo e benefício de um determinado item de informação. Uma forma de estimar o custo de uma mensagem, por exemplo, é observar o tamanho do texto contido, geralmente um dos primeiros fatores usados pelo destinatário para estimar o custo de uma mensagem. O custo de uma informação pode ser também relacionado à sua produção. Em uma mensagem, quando o número de destinatários é muito grande (i.e. lista de mensagens e mala direta), o custo do remetente por destinatário será baixo, enquanto que uma mensagem produzida para um único destinatário tem um custo maior.

Filtragem Colaborativa ou *Filtragem Social* filtra informação baseada no julgamento individual da qualidade de um item compartilhado em uma comunidade de usuários. Nesse tipo de filtragem, a relação entre usuários é a parte central na filtragem. Em uma comunidade é observado que, para um dado usuário alvo, existem usuários com perfil mais similares a ele do que outros. Essa relação de semelhança quanto à preferência ou necessidade de informação pode ser aproveitada para que, através do compartilhamento da preferência sobre a coleção de informações, seja possível fazer a filtragem, ou seja, a relação entre usuários pode ser usada na recomendação cruzada de itens entre usuários semelhantes. Um exemplo de filtragem colaborativa é a seleção de mensagens baseado na relação do remetente com o destinatário. As mensagens enviadas por colaboradores do mesmo projeto que se está trabalhando ou por superiores são mais relevantes do que mensagens de um remetente externo desconhecido.

Tradicionalmente os sistemas de recomendação personalizados se encaixam em duas categorias de filtragem: colaborativa e baseada em conteúdo. Diversos

sistemas de filtragem de informação foram construídos utilizando uma dessas técnicas de filtragem ou mesmo uma combinação de ambas. As características dessas técnicas de filtragem, bem como suas vantagens e desvantagens, são discutidas com mais detalhes nas seções seguintes.

3.2.2 Filtragem Baseada em Conteúdo

Nos sistemas de filtragem baseado em conteúdo, utilizamos apenas as preferências do próprio usuário. O foco desse tipo de sistema está em aprender o perfil do usuário a partir da descrição dos itens que ele avalia, para que possa produzir futuras recomendações de itens similares ao perfil aprendido.

A descrição da informação será relevante nesse tipo de filtragem. Um filme, por exemplo, pode ser descrito por um conjunto de atributos como mostrado na figura 3.1:

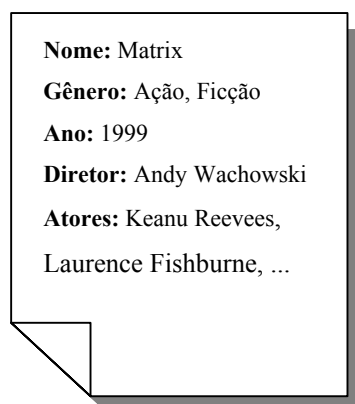


Figura 3.1 – Exemplo da descrição de um filme

As preferências de um usuário são representadas por suas avaliações sobre os itens de informação. O objetivo da filtragem baseada em conteúdo é aprender sobre as características da descrição dos itens de informação relevantes e não relevantes para um usuário, para que possa fazer um julgamento da relevância de novos itens de acordo com as suas preferências. A figura 3.2 ilustra o problema da filtragem baseada em conteúdo para o caso dos filmes:

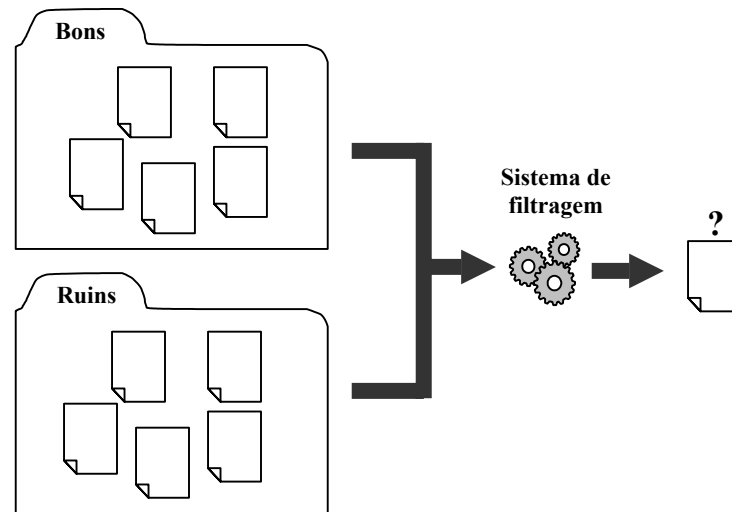


Figura 3.2 – Problema de filtragem baseada em conteúdo

Na figura 3.2, o sistema de filtragem baseado em conteúdo conhece a avaliação do usuário para alguns filmes. A partir desta avaliação e da descrição desses filmes, o sistema deve ser capaz de predizer se o usuário irá gostar do novo filme baseando-se na sua descrição. Algoritmos de aprendizagem são usados para derivar classificadores capazes de predizer quais itens um usuário irá gostar a partir dos atributos [Pazzani 97], ou são inferidas similaridades entre os itens a partir do conhecimento sobre o domínio do problema tornando a filtragem uma aplicação de CBR [Burke 99]. A preferência do usuário freqüentemente é usada também para construir um protótipo contendo indicadores do interesse do usuário sobre determinado tópico, geralmente representados através de um conjunto de palavras-chaves e pesos associados à relevância da palavra [Cotter 00], [Krukwich 96], [Balabanovic 97].

Os sistemas que utilizam filtragem baseada em conteúdo possuem algumas desvantagens que são próprias desse tipo de filtragem. Uma das principais desvantagens está na dependência quanto à representação do conteúdo dos itens. Em alguns casos, o tipo de informação filtrada não pode ser representada de forma satisfatória através de atributos como, por exemplo, objetos multimídia como sons, vídeos e figuras que são de difícil extração de características. Em outros casos, a representação das características através de atributos, ainda que possível, pode

omitir características que são relevantes para os usuários no momento de escolher um item. Por exemplo, quando a informação filtrada é sobre um filme, pode não ser o bastante saber sobre atores, gênero e diretor, atributos frequentemente usados para descrever um filme, para determinar sua qualidade, enquanto que características como atuação, trilha sonora ou fotografia podem ser determinantes.

Outro problema relacionado à sistemas de filtragem de conteúdo está na super especialização do sistema em tópicos frequentes no perfil do usuário. Quando o sistema aprende sobre o perfil do usuário, tende a recomendar apenas itens muito similares àqueles que o usuário já avaliou no passado, tornando as recomendações pouco diversificada.

Por último, a filtragem de conteúdo possui uma forte dependência em relação a quantidade de avaliações realizadas pelo usuário, uma vez que a única influência na qualidade das recomendações são estas avaliações.

3.2.3 Filtragem Colaborativa

Nos sistemas de filtragem colaborativa há a formação de uma comunidade de usuários que interagem com o sistema fornecendo avaliações. Membros da comunidade podem usufruir da experiência de outros ao mesmo tempo em que fornecem informações sobre sua experiência.

Sistemas colaborativos têm sua ênfase em algoritmos para produzir recomendações baseando-se em usuários com perfis semelhantes. Em vez de recomendar itens similares aos avaliados pelo usuário, são recomendados itens cujos usuários similares tenham gostado. A recomendação é baseada na similaridade entre usuários e não mais entre itens. As técnicas para filtragem colaborativa são discutidas em [Breese 98] e [Herlocker 00] e serão aprofundadas no capítulo 5.

A principal vantagem da filtragem colaborativa é não possuir vários dos problemas inerentes à filtragem baseada em conteúdo. Como toda a avaliação usada na recomendação é baseada no julgamento humano, não existe mais o problema de dependência da representação do conteúdo dos itens, já que não é necessário para o

sistema conhecer o conteúdo para análise. De fato, tudo que é preciso para fazer a filtragem colaborativa é um identificador único para cada item e para cada usuário, além da relação das avaliações dos usuários para os itens.

As avaliações feitas por humanos também têm a vantagem de levar em consideração características realmente relevantes para a avaliação de qualidade, inclusive as que não são representáveis facilmente em computadores. Além disso, na filtragem colaborativa, a qualidade das recomendações não é mais dependente apenas das avaliações do próprio usuário, mas também dos membros da comunidade, o que permite ter recomendações de qualidade mesmo com poucas avaliações e torna as recomendações mais diversificadas.

A filtragem colaborativa possui também alguns problemas próprios. Um desses problemas está relacionado a novos itens dentro da base de informação. Logo que um novo item é apresentado ao sistema, este será incapaz de recomendá-lo até que este seja avaliado. Isso se dá porque na filtragem colaborativa o sistema depende da experiência dos membros da comunidade para determinar a relevância de um item.

O sucesso de um sistema de filtragem colaborativa é dependente da comunidade de usuários e do tipo de informação que se deseja filtrar. Quanto maior a comunidade e a quantidade de avaliações que seus membros fornecem, melhor será a qualidade das recomendações. Também é importante observar a base de informação. Se esta base for pequena comparada à comunidade e crescer lentamente, um sistema colaborativo será capaz de gerar boas recomendações. Entretanto se ela for muito maior do que a comunidade ou crescer a uma velocidade muito grande, é provável que um sistema de recomendações puramente colaborativo gere recomendações imprecisas. Uma solução é gerar recomendações baseando-se numa combinação de filtragem colaborativa com filtragem baseada em conteúdo.

3.2.4 Filtragem Híbrida

Das técnicas de filtragem de informação mais usadas, nenhuma parece resolver por si só a necessidade de filtragem de informação que se requer em um sistema de

recomendação. Um sistema de recomendação precisa ser capaz de gerar recomendações variadas e de qualidade, aprender rapidamente sobre o perfil do usuário sem exigir muito tempo deste para fornecer suas preferências e ainda filtrar novas informações assim que inseridas no sistema com base nas características relevantes para o usuário.

A filtragem baseada em conteúdo apresenta características desejáveis em um sistema de recomendação, mas também desvantagens, como a filtragem baseada na representação interna da informação, dependência da quantidade de avaliações realizadas pelo usuário e geração de recomendações pouco variadas. Filtragem colaborativa, por sua vez, não possui essas desvantagens, entretanto possui características indesejáveis num sistema, como incapacidade de filtrar novas informações e a dependência do tamanho da comunidade.

As duas técnicas de filtragem, no entanto, possuem características complementares que, quando combinadas, podem ser capazes de suprir as necessidades de filtragem em um sistema de recomendação. Surge então a idéia de combinar as duas metodologias para formar um sistema híbrido, que possua as vantagens dos dois modelos de filtragem suprimindo as desvantagens de cada tipo individualmente. Um dos primeiros sistemas de filtragem híbrida, o FAB [Balabanovic 97] da Stanford University, filtra páginas da Internet usando agentes de filtragem que usam metodologias de híbrida, com filtragem de conteúdo e colaborativa. Nesse sistema, cada usuário possui um *Agente de Seleção* que possui o perfil do usuário formado por palavras extraídas dos documentos de sua preferência. Este agente é responsável por buscar recomendações para o usuário que representa junto aos *Agentes de Coleta*, que buscam documentos na *Web* por meio da filtragem de conteúdo, ou junto a outros *Agentes de Seleção* cujo perfil do usuário se assemelha.

GroupLens [Good 99] é um sistema híbrido de filtragem de filmes, que utiliza agentes dentro da comunidade em um sistema de filtragem colaborativa. Os agentes chamados *filterbots* são programados para avaliar os filmes segundo o conteúdo

destes. Estes agentes são tratados pelo sistema de filtragem colaborativa como usuários comuns que fazem muitas avaliações. Os *filterbots* mostraram-se capazes de melhorar a performance da filtragem agindo no sistema como usuários comuns, que se associam aos usuários reais na filtragem colaborativa. Indiretamente, por meio desses agentes, o sistema executa também uma filtragem de conteúdo. Outros utilizam filtragem híbrida para filtrar programas de TV [Cotter 00] e notícias [Claypool 99], apresentando resultados satisfatórios.

3.3 Medindo a qualidade das recomendações

A avaliação de sistemas de recomendações é ainda uma questão aberta no estudo desses sistemas [Herlocker 00]. Isso se deve principalmente a dois fatores. Primeiro, o desenvolvimento de sistemas de recomendação é uma inovação recente e ainda não há um consenso sobre as métricas adequadas para avaliar esse tipo de sistema. O segundo fator está associado ao entendimento de qual é a tarefa de um sistema de recomendação, ou seja, que tipo de recomendação o sistema deve fornecer ao usuário. Nesta seção são analisadas as tarefas que os sistemas de recomendação podem exercer. Uma tarefa define o que é desejado em uma recomendação e será importante na determinação de que métrica usar para avaliar a performance de um sistema.

3.3.1 Tarefas de um Sistema de Recomendação

Identificar as tarefas de um sistema é o primeiro passo para saber como avaliá-lo de forma apropriada. A tarefa do sistema descreve a natureza da comunicação com os usuários e o objetivo do usuário quando utiliza o sistema. A escolha da métrica apropriada para avaliar um sistema depende da tarefa identificada para o sistema. Em sua tese, Herlocker [Herlocker 00] lista seis tarefas possíveis para um sistema de recomendação. As tarefas que mostramos aqui são um resumo das sugeridas nesse trabalho, divididas em três tarefas mais gerais que julgamos apropriadas:

TAREFA 1: Selecionar os itens relevantes para o usuário.

Nesse caso o usuário deseja receber recomendação apenas dos itens que se assemelham com o seu perfil para que possa escolher quais itens consumir. Para o sistema cabe diferenciar e selecionar estes itens, mostrando ao usuário apenas os relevantes.

Se o usuário deseja manter uma atenção constante sobre a informação, e quer que o sistema o notifique de qualquer item relevante que surja no sistema, então a tarefa de seleção de itens relevantes é apropriada para o sistema. Um exemplo disso seria um sistema de recomendação de artigos. Uma vez conhecido o perfil do pesquisador, um sistema poderia automaticamente informá-lo sobre novas publicações de seu interesse.

Quando o usuário tem como finalidade obter uma visão dos itens, tendo os itens não relevantes escondidos e os relevantes mostrados, a tarefa do sistema também será a seleção de itens. Um exemplo disso seria um sistema de recomendação de notícias. Uma vez conhecido o perfil do usuário, o sistema poderia apresentá-lo um jornal apenas com as notícias relevantes segundo seu interesse.

Nessa tarefa, as recomendações dependerão do número de itens relevantes que existem na base de informação.

TAREFA 2: Ordenar os itens segundo sua relevância para o usuário.

Nesse caso o usuário deseja obter uma lista de itens ordenados segundo sua relevância. O usuário pode então navegar pelos itens na ordem do mais para o menos relevante na busca de um item para consumir. Por exemplo, em um sistema de recomendação de filmes, uma lista de filmes ordenados em ordem crescente de relevância poderia ser mostrado ao usuário, e este percorreria a lista nessa ordem até encontrar um filme que não tenha assistido.

Para visualizar quais itens são mais relevantes, o sistema também deverá ordenar os itens pela sua relevância e mostrar apenas os mais similares ao perfil do

usuário. Por exemplo, no sistema de recomendação de filmes o sistema poderia retornar uma lista dos 10 filmes mais interessantes para o usuário.

Essa tarefa independe da existência de itens relevantes na base de informação. A tarefa do sistema será ordenar corretamente os itens pela sua relevância para o usuário.

TAREFA 3: Prever a avaliação dos itens para um usuário.

Nesse caso o usuário deseja receber uma previsão de qual seria sua avaliação para um item que ele ainda não avaliou.

Quando o usuário analisa uma seqüência de itens selecionando os relevantes e eliminando os não relevantes, mas não deseja que o sistema omita nenhum deles, ele poderá ser auxiliado por uma avaliação prévia de cada item. Como, por exemplo, em um sistema de mensagens eletrônicas poderia ser associado a cada e-mail um escore de relevância. Nesta situação o usuário não deseja que sua mensagens sejam escondidas dele, mas que possa ser auxiliado no momento de decidir qual e-mail abrir e qual apagar.

O usuário pode também pedir avaliação para um item específico para decidir se vale a pena consumi-lo. Por exemplo, em um sistema de recomendação de filmes, ele poderia pedir a previsão de sua avaliação de um filme que está em cartaz e tomar a decisão de se irá assisti-lo ou não de acordo com essa previsão.

Na maior parte dos casos, os sistemas de recomendação utilizam algoritmos de previsão de escore para executar qualquer uma dessas tarefa. Se o sistema puder prever com precisão os escores de cada item, o que é feito para executar a tarefa 3, ele poderá usar esse escore tanto para selecionar os itens relevantes, no caso da tarefa 1, como para ordenar os itens pela sua relevância, no caso da tarefa 2. Entretanto, a tarefa 3 propriamente não é comum em sistemas de recomendação. A maior parte dos sistemas preocupam-se mais em selecionar os itens relevantes ou em gerar uma lista de itens ordenados pela sua relevância. Para esta dissertação foram consideradas as tarefas 1 e 2 pois julgamos essas as mais relevantes para a avaliação

de um sistema de recomendação. A seguir serão discutidas as métricas utilizadas para avaliar a capacidade do sistema de realizar essas duas tarefas e porque são mais apropriadas para essas tarefas.

3.3.2 Métricas

As métricas usadas nas avaliações feitas nesse trabalho são apresentadas nesta seção, onde também especificamos para qual tarefa é mais apropriada.

Área sob a curva ROC

A curva “*relative operating characteristic*”, ou simplesmente curva ROC, foi introduzida na comunidade de RI por Swets [Swets 63]. A curva ROC é utilizada para medir o quanto um valor produzido por um sistema é capaz de distinguir os elementos relevantes dos não relevantes.

O sistema medido tem como saída uma variável de relevância associada a cada elemento. O conhecimento sobre a relevância dos elemento permite construir duas curvas de distribuição, uma para os valores obtidos para os elementos relevantes e outra para os elementos não relevantes. No momento de selecionar os elementos relevantes, o sistema usa um limiar t . Os elementos que ultrapassem esse limiar serão considerados relevantes e portanto selecionados. Caso contrário, esse elementos serão considerados irrelevantes e então rejeitados. Para cada valor escolhido para o limiar t é possível calcular a *cobertura* (proporção dos elementos relevantes que são selecionados) e o *ruído* (proporção dos elementos não relevantes que são selecionados). A figura 3.3 ilustra as distribuições dos elementos relevantes e não relevantes e o acerto e erro para um limiar t escolhido.

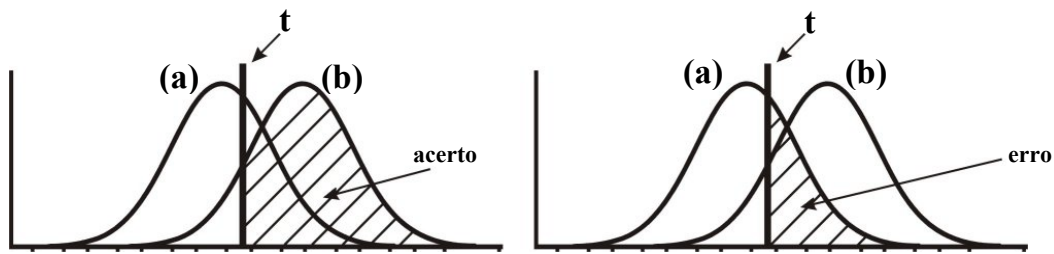


Figura 3.3 – Curvas da distribuição dos valores para elementos relevantes (b) e não relevantes (a). O limiar t determina a precisão (taxa de acerto) e o ruído (taxa de erro) do sistema.

A cobertura será a razão do número de acertos pelo número de elementos relevantes e o ruído a razão do número de erros pelo número de elementos não relevantes.

A utilização da cobertura e do ruído é bastante comum na avaliação de sistemas que tem como objetivo a seleção binária através de uma variável resultante separada por um limiar. O problema é que os valores dessas grandezas dependem do limiar escolhido. A curva ROC por sua vez é capaz de avaliar a capacidade do sistema separar os elementos relevantes dos não relevantes sem depender da escolha de um limiar, permitindo comparar dois sistemas mais facilmente. A curva ROC é a curva obtida quando desenhado em um plano cartesiano os valores da cobertura (ordenada) versus o ruído (abscissa) para diferentes valores do limiar t . A figura 3.4 mostra um gráfico com uma curva ROC e alguns pontos para diferentes valores de t .

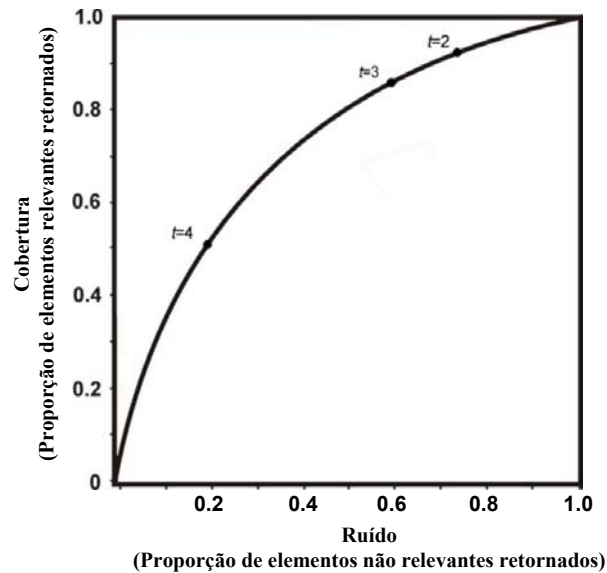


Figura 3.4 – Exemplo de uma curva ROC. Os pontos t indicados na curva representam diferentes limiares usados para separar itens relevantes de não relevantes para um sistema com sinal de saída entre 1 e 5. Por exemplo, se o limiar usado for de $t=3$ a cobertura será igual a 0,85 enquanto o ruído será igual a 0,6.

As curvas ROC são bastante úteis para observar a capacidade de um algoritmo de FI separar informação relevante da não relevante. Entretanto, comparar várias curvas ROC pode ser um processo trabalhoso e impreciso. A área sob uma curva ROC pode ser usada como um valor que expressa a capacidade do sistema discriminar os itens relevantes dos não relevantes. A figura 3.5 mostra o exemplo de várias curvas ROC e a área A em cada situação.

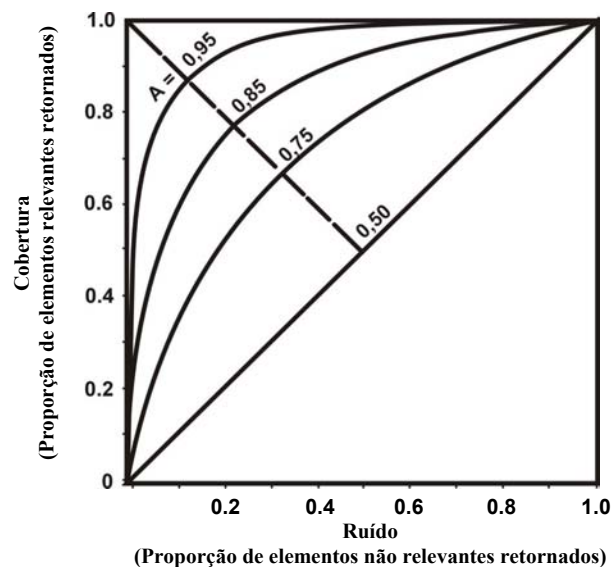


Figura 3.5 – Exemplos de curvas ROC da cobertura vs. ruído. A área A sob a curva indica a separação entre os elementos relevantes dos não relevantes.

Em [Hanley 82] é mostrado que a área sob a curva ROC representa a probabilidade de que, dados dois itens escolhidos aleatoriamente, sendo um relevante e outro não, o sistema distinguirá corretamente os dois elementos. É fácil concluir que, se a área sob a curva é de 0,5, então em 50% das vezes o sistema distinguiria corretamente os elementos e em 50% das vezes erroneamente, o que equivaleria a uma classificação puramente aleatória, ou seja, a um desempenho nada satisfatório. Valores menores que 0.5 indicam que o sistema está trocando os elementos relevantes pelos não relevantes, enquanto valores próximos de 1.0 indicam uma boa precisão na seleção dos elementos.

Como a área sob a curva ROC é uma métrica de seleção, i.e. mede a capacidade do sistema selecionar os itens relevantes, ela será mais apropriada para a tarefa 1 da seção 3.3.1. Descobrir quais são os itens relevantes ainda dependerá de um limiar que pode ser calculado para o sistema ou para cada usuário, mas o valor da área sob a curva ROC é capaz de diferenciar a capacidade de seleção de dois algoritmos de filtragem, independente do limiar.

A área sob a curva ROC não é apropriada para a tarefa 2 pois não leva em consideração a ordem gerada pela predição do sistema. Também não é apropriado para a tarefa 3, pois não mede a precisão do sistema na predição de cada item individualmente.

Breese

A métrica criada por Breese et al. [Breese 98] é usada para medir a utilidade de uma lista ordenada por um sistema de filtragem para o usuário. Em um sistema de filtragem de informação onde o resultado de uma filtragem é retornado como uma lista de itens ordenados pela relevância, é presumível que o usuário irá investigar apenas os primeiros elementos da lista na sua busca por itens que lhe interessam. A métrica Breese procura medir a utilidade de uma lista levando em consideração a tendência do usuário de observar apenas os primeiros itens da lista.

Para cada item de uma lista, sua utilidade será a diferença entre a avaliação dada pelo usuário e o valor médio d no intervalo de avaliação. Para a constante d geralmente se assume um valor no intervalo de avaliação que indica neutralidade em relação à relevância de um item. Na lista, cada item terá sucessivamente uma probabilidade menor de ser visto pelo usuário. A constante α , chamada de *half-life*, é a posição do item na ordem gerada pelo sistema que terá 50% de chance de ser observada. A utilidade de uma lista ordenada para o usuário a é então:

$$R_a = \sum_j \frac{\max(r_{a,j} - d, 0)}{2^{(j-1)/(\alpha-1)}} \quad (\text{Eq. 3.1})$$

Onde j é a posição do item na lista ordenada pelo sistema e $r_{a,j}$ é a avaliação feita pelo usuário ao item que ocupa a posição j . Breese utiliza um *half-life* α igual a 5 e acrescenta que a utilização de um valor igual a 10 apresentou pouca diferença nos resultados. Assumimos um *half-life* igual a 5 indicado na literatura como apropriado para avaliar as recomendações com a mesma base de dados utilizada neste trabalho [Breese 98]. O valor final da métrica para o conjunto de usuários testados é calculado como segue :

$$R = 100 \frac{\sum_a R_a}{\sum_a R_{\max}} \quad (\text{Eq. 3.2})$$

Onde R_{\max} é a utilidade máxima obtida quando todos os itens são ordenados de acordo com a avaliação do usuário, ou seja, quando a ordenação dos itens pelo sistema é igual à ordenação pelas avaliações do usuário. O valor que R_a pode assumir depende do número de itens retornados em uma lista e da distribuição das avaliações do usuário. Essa última transformação permite considerar os resultados independente desses fatores, pois o valor da métrica é normalizado através da divisão pelo valor máximo de utilidade de uma lista.

A métrica criada por Breese é apropriada para avaliar a tarefa 2 por medir a capacidade do sistema de ordenar os itens de acordo com sua relevância para o usuário e também por dar mais ênfase à precisão dos primeiros itens na ordenação do sistema. Ela já não é apropriada para a tarefa 1, pois não computa a capacidade de selecionar os itens relevantes dos não relevantes, mas sim a capacidade de ordenar os itens pela sua relevância para o usuário. Também não é apropriado para a tarefa 3, pois não mede a precisão da predição da avaliação de cada item.

Outras métricas

Além das métricas apresentadas aqui existem outras métricas que poderiam ser utilizadas para medir a qualidade das recomendações. Métricas como precisão e cobertura, correlações de valores como Pearson ou de ordens como a Spearsman e Kendall Tau e a *npdm* [Yao 95], são outras opções de uma vasta seleção de métricas já usadas para avaliar sistemas de filtragem de informação. Todas as métricas possuem vantagens e significados diferentes. As métricas ROC e Breese escolhidas para este trabalho formam um conjunto abrangente na avaliação de um sistema de filtragem com relação às possíveis tarefas de um sistema de recomendação. Um estudo mais aprofundado das métricas que serviram também como base para essa escolha encontra-se em [Herlocker 00].

3.4 Conclusão

Esse capítulo teve como objetivo apresentar o estado da arte no desenvolvimento de sistemas de recomendação. Os sistemas não personalizados foram apresentados como aqueles que dão suporte às recomendações, fazendo apenas o papel de coleta de preferência e distribuição de recomendações. Os sistemas personalizados por sua vez identificam e aprendem as preferências e necessidades de um usuário, gerando recomendações apropriadas ao perfil do usuário. As metodologias de filtragem de informação utilizadas em sistemas personalizados são apresentadas com ênfase para a filtragem colaborativa e a

baseada em conteúdo. As vantagens e desvantagens das duas metodologias de filtragem são discutidas e as características complementares das metodologias são destacadas como vantagens da combinação das duas em sistemas de filtragem híbrida.

Por último discutimos as formas de medir a qualidade das recomendações produzidas por um sistema de recomendação. Uma divisão das tarefas possíveis de um sistema de recomendação é proposta para orientar na decisão de como avaliar a performance de um sistema da melhor forma. As métricas ROC [Hanley 82] e Breese [Breese 98] foram selecionadas por serem mais apropriadas para avaliar as tarefas de seleção da informação relevante e de ordenação da informação pela sua relevância respectivamente.

Capítulo 4

Coletando as preferências do usuário

Para possibilitar a aprendizagem da preferência de um usuário é necessário que se possa coletar informação sobre sua preferência. Sistemas de recomendação utilizam diferentes técnicas para determinar a preferência do usuário sobre um item ou pedaço de informação. O sistema armazena a preferência do usuário através de um valor que quantifica o interesse sobre a informação. Esse valor pode ser obtido com a participação ativa do usuário através da marcação de uma nota (um valor em um intervalo discreto) a um item ou informação. Nesse caso onde o usuário indica diretamente sua preferência efetua-se uma *avaliação explícita*. Outra forma de se obter informação sobre a preferência de um usuário é indiretamente, através da observação do uso que o usuário faz do sistema, chamada *avaliação implícita*.

4.1 Indicadores Implícitos

Em [Nichols 97] são discutidas diferentes formas de obter-se indicadores de preferência implícitos e as vantagens e desvantagens das avaliações implícitas. Em [Oard 98] é exposta uma divisão dos tipos de avaliações implícitas identificadas em [Nichols 97] em três categorias: *análise*, *retenção* e *referência*.

Análise refere-se ao comportamento de um usuário ao examinar um item ou uma informação. Selecionar um item de uma lista para obter mais informações sobre este é um tipo de indicação de preferência implícita de análise. Em [Morita 94] e [Konstan 97] é indicada uma correlação entre o tempo de leitura de artigos de grupos de notícia e a preferência explícita de um usuário, dando indícios de que o tempo de análise pode ser um indicador de preferência implícito. Hill et al. [Hill 92] desenvolve a idéia de análise baseado no uso da informação, no que chama de “uso de leitura” e “uso de edição”, em uma analogia com a idéia de que a utilização de um utensílios físicos deixa marcas que podem ser indicadores de relevância ou preferência, como em um livro de uma biblioteca bastante lido apresentando marcas na capa e páginas. Em situações como navegar por um texto digital, ao deslizar mais lentamente ou com mais frequência por uma parte do texto, tem-se um indicativo de preferência dessa parte. Por fim, um indicador de preferência por análise muito

comum em muitos sistemas de comércio eletrônico é a compra de produtos, que em muitos casos são utilizados para reconhecer a preferência do usuário e recomendar novas compras.

Retenção refere-se a ações nas quais o usuário dá indicações de interesse em fazer mais uso de determinado item. Ações como salvar um item ou uma referência para esse item podem ser entendidos como indicadores de preferência. Em [Rucker 97] é desenvolvido um sistema de navegação personalizado onde os indicadores de preferência são obtidos na lista de endereços salvos no navegador do usuário. Em um sistema de gerência de e-mail, salvar uma mensagem em uma pasta separada pode ser entendido como indicador da preferência do usuário. Outros indicadores implícitos de preferência são imprimir ou apagar um item, sendo este último um indicador negativo de preferência.

Referência é um indicador de preferência visto como uma ligação entre dois itens que pode ser associado ao interesse do produtor ou do consumidor dessa referência pela informação contida nos itens. Passar adiante uma mensagem produz uma referência entre as duas mensagens, a original e a repassada, e uma indicação de preferência do remetente. No sistema Tapestry [Goldberg 92] os usuários podem requisitar as mensagens do seu grupo as quais um participante repassou ou respondeu. Referências dentro de um hipertexto ou de um texto comum também podem ser utilizadas como indicadores de preferência usados com frequência em sistemas de RI para melhorar a qualidade da informação retornada, como no algoritmo *PageRank* [Page 98] usado no Google (www.google.com). O sistema PHOAKS [Hill 92] minera as mensagens postadas para grupos de notícia buscando referências a páginas na *Web* e cria lista de páginas mais populares em cada grupo de notícias.

A principal vantagem da avaliação implícita é a ausência do custo adicional para o usuário de avaliar os itens, uma vez que o sistema colhe a sua preferência observando apenas suas ações dentro do sistema. Potencialmente qualquer iteração

do usuário com o sistema pode ser utilizada como indicador de preferência [Claypool 01].

4.2 Indicadores explícitos

Indicadores explícitos fornecem uma indicação clara e direta da preferência de um usuário sobre um item. Os sistemas que possuem esse tipo de avaliação geralmente disponibilizam em sua interface componentes que permitem a avaliação de cada item mostrado. O objetivo é facilitar ao máximo a avaliação do usuário para que em muitos casos um simples clique seja suficiente para expressar sua preferência sobre um item.

Muitos dos sistemas de recomendação utilizam esse tipo de avaliação como são os casos dos sistemas PTV [Cotter 00], Ringo [Shardanand 95], Bellcore Video Recommender [Hill 95], FilmConseil [Perny 02] e MovieLens [Good 99].

[5] = Must See [4] = Will Enjoy It [3] = It's OK [2] = Fairly Bad [1] = Awful		
YOUR RATING	GENRE	TITLE
? unseen	Adventure, Fantasy	Lord of the Rings: The Fellowship of the Ring, The (2001)
? unseen	Adventure	Himalaya (Himalaya - L'enfance D'un Chef) (1999)
4	Adventure, Drama	Cast Away (2000)
3	Action, Adventure Children's, Comedy	Spy Kids (2001)
? unseen	Action, Adventure Drama	Left Behind: The Movie (2001)
2	Adventure, Children's Fantasy	Harry Potter and the Sorcerer's Stone (2001)
3	Adventure, Comedy	O Brother, Where Art Thou? (2000)
4	Crime	
5	Adventure, Animation Fantasy, Sci-Fi	Final Fantasy: The Spirits Within (2001)
3	Adventure, Drama Fantasy, Sci-Fi	K-PAX (2001)
2	Adventure, Animation Sci-Fi	Titan A.E. (2000)

Submit ratings and see more

Figura 4.1 – Tela do sistema MovieLens. O usuário utiliza uma caixa de opções para avaliar um filme com um escore entre 1 e 5.

A figura 4.1 mostra uma tela do sistema MovieLens de recomendação de filmes. Para cada filme mostrado o usuário utiliza uma caixa de opções para indicar a sua avaliação para o respectivo filme.

4.3 Indicadores implícitos versus indicadores explícitos

A necessidade de se conhecer o perfil do usuário para a execução da filtragem da informação leva ao questionamento de qual é a melhor forma de obter indicadores de preferência de um usuário. Atualmente, as duas formas de obter tais indicadores, implicitamente e explicitamente, apresentam vantagens e desvantagens.

A forma com que a preferência do usuário é coletada será determinante para o sucesso de um sistema de recomendação. Alguns trabalhos mostram que as avaliações implícitas são indicadores relevantes sobre a preferência do usuário [Morita 94], [Konstan 97] e [Claypool 01]. Entretanto, Nichols [Nichols 97] comenta que as avaliações implícitas possuem um custo computacional maior do que as avaliações explícitas e que existem limitações quanto ao transporte e armazenamento dos indicadores implícitos que podem ser em grande quantidade. Claypool et al [Claypool 01] afirma que esses indicadores também possuem menor “valor” do que uma avaliação explícita, mas que, apesar disso, é possível combinar vários indicadores implícitos para obter um indicador mais preciso da preferência do usuário. Entretanto, isso acarretaria um custo ainda maior para captar esses indicadores, transportar e armazenar. Os indicadores implícitos ainda são pouco utilizados e grande parte dos sistemas de recomendação são baseados principalmente em avaliações explícitas.

Quando a avaliação explícita é utilizada, em geral é necessário um número importante de avaliações e com isso corre-se o risco de desinteressar o usuário, afastando-o do sistema ou fazendo com que atue passivamente não fornecendo avaliações. Uma possível solução seria tentar incentivar o usuário a fazer avaliações provendo algum benefício relacionado à quantidade de avaliações que ele efetuou. Avery et al [Avery 99] fizeram uma análise de sistemas de incentivo a partir da teoria dos jogos no intuito de encorajar os usuários a fornecer avaliações. Entretanto, na avaliação explícita existe não apenas a necessidade de motivar o usuário a fornecer avaliações, mas também a necessidade de aprender o máximo possível sobre um usuário com cada avaliação fornecida por ele.

4.4 Aprendizagem Ativa

Para muitos algoritmos de aprendizagem, é comum assumir que existe um grande conjunto de exemplos classificados para o treinamento. Entretanto, em alguns problemas de aprendizagem, a classificação de exemplos é um processo caro, que em alguns casos requer trabalho manual. Nesses casos é desejável utilizar um número reduzido de exemplos de treinamento sem comprometer a precisão do classificador. *Aprendizagem ativa* é o paradigma onde os algoritmos de aprendizagem possuem algum controle sobre as entradas a partir das quais é realizado o treinamento.

Algoritmos de aprendizagem ativa são divididos em dois paradigmas: *membership queries* e *selective sampling*. No paradigma de *membership queries* [Angluin 88] é possível para o algoritmo construir exemplos artificiais e depois pedir sua classificação. O algoritmo constrói exemplos cuja classificação é importante para formular uma hipótese consistente rapidamente, ou seja, com um número reduzido de exemplos. Por exemplo, em um problema de reconhecimento de imagens, um algoritmo de aprendizagem poderia gerar uma imagem com características diferente das já apresentadas, e requisitar de um supervisor humano a classificação da imagem. O problema dessa metodologia está na possibilidade de levar a exemplos mal construídos ou não significativos [Lindenbaum 99], como por exemplo, no problema de reconhecimento de imagens apresentar uma imagem confusa.

O paradigma de *selective sampling* é mais restritivo do que o paradigma de *membership queries*. O problema consiste em, dentre um conjunto de exemplos cuja classificação é desconhecida, selecionar o próximo exemplo que será classificado pelo supervisor e usado no treinamento. A tarefa do algoritmo de *selective sampling* consiste em selecionar os exemplos cuja classificação é importante para formular uma hipótese consistente rapidamente, ou seja, com a classificação do menor número de exemplos possível. O processo de aprendizagem com *selective sampling* pode ser descrito como um procedimento iterativo em que, a cada iteração, o

algoritmo de *selective sampling* é chamado para selecionar um exemplo não classificado e o supervisor é chamado para classificar o exemplo. O novo exemplo é usado no treinamento para atualizar o classificador. O procedimento é repetido até atingir um critério de parada, geralmente quando o classificador atinge a performance desejável na classificação de exemplos em um conjunto de teste separado.

Vários algoritmos de aprendizagem ativa foram desenvolvidos para acelerar o aprendizado de diversos algoritmos de classificação, como Redes Neurais [Cohn 94], algoritmo C4.5 de indução de regras [Lewis 94], *Support Vector Machine* [Tong 00] e KNN [Hasenjager 98] [Lindenbaum 99].

4.5 Aprendizagem Ativa em Sistemas de FI

Em um sistema de filtragem de informação, cada usuário dispõe de um conjunto de exemplos sobre as suas preferências, i.e. avaliações de itens, e a tarefa do sistema de filtragem é aprender o perfil de cada usuário para fornecer predições de itens ainda não avaliados. Como cada usuário possui características próprias, não é possível aprender um conceito geral que possa ser aplicado para todos usuários. Cada usuário possui suas preferências individuais e essas preferências devem ser aprendidas apenas através dos exemplos (avaliações) fornecidos por ele. Portanto cada usuário é uma instância de um problema de aprendizagem.

Quando a avaliação explícita é utilizada, é delegada ao usuário a tarefa de produzir manualmente avaliações expressando sua opinião sobre os itens. Nesse caso, a produção de um exemplo é uma tarefa cara e é desejável reduzir o número de exemplos de treinamento necessários e ao mesmo tempo manter a qualidade de futuras predições. Dessa forma seria possível diminuir a necessidade de avaliar cada vez mais itens para melhorar a qualidade das predições. Através da requisição da avaliação do usuário a itens específicos, aqueles que ajudam a aprender mais rapidamente os aspectos mais relevantes relacionados ao seu perfil, o sistema poderia conduzi-lo a fornecer suas preferências com uma quantidade menor de

avaliações. Requer-se então que seja desenvolvida uma metodologia capaz de fazer essa aprendizagem ativa no contexto de um sistema de filtragem de informação.

Um sistema de filtragem normalmente interage de forma passiva com o usuário, apenas recebendo as avaliações dos itens que o usuário escolhe para avaliar. O modelo de avaliação de um sistema de filtragem passivo é mostrado na figura 4.2.

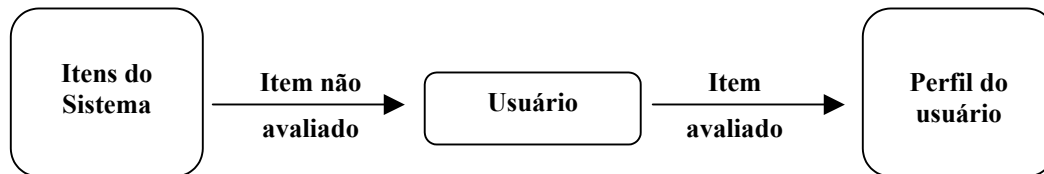


Figura 4.2 – Modelo de avaliação de uma sistema de filtragem passivo

Nesse modelo o usuário avalia os itens do sistema em uma ordem indefinida. Os itens avaliados pelo usuário serão determinados por diversos fatores dos quais o sistema não tem controle. O usuário simplesmente avalia itens que surgem no sistema ao longo de sua utilização. A forma que o usuário irá navegar pelo sistema, quais itens serão mostrados para ele e quais serão avaliados não podem ser determinados. Esse modelo de avaliação é usado nos sistemas de recomendação atuais.

Neste trabalho propomos uma metodologia de aprendizagem ativa para filtragem colaborativa. Para isso, o paradigma de *selective sampling* é a escolha mais apropriada devido as características do problema da filtragem, no qual o conjunto de itens para os quais o usuário pode fornecer avaliações é determinado *a priori*. Além disso, a alternativa de *membership queries* representa um problema bem mais complexo, pois envolve a construção de itens artificiais que devem fazer sentido para o usuário.

A tarefa da *selective sampling* em filtragem de informação é selecionar os itens para os quais será requisitada a avaliação do usuário com os seguintes objetivos:

1. minimizar a quantidade de avaliações do usuário necessária para alcançar uma qualidade específica de recomendações.

2. maximizar a qualidade das recomendações para uma quantidade fixa de avaliações do usuário.

Como alternativa ao modelo de avaliação mostrado na figura 4.2, o modelo de avaliação de um sistema de filtragem ativo seleciona os itens que um usuário avalia. Para isso um algoritmo *selective sampling* seleciona dentre os itens do sistema aqueles que ajudam a aprender mais rapidamente sobre o perfil do usuário. A figura 4.3 ilustra o modelo de avaliação de um sistema de filtragem ativo.

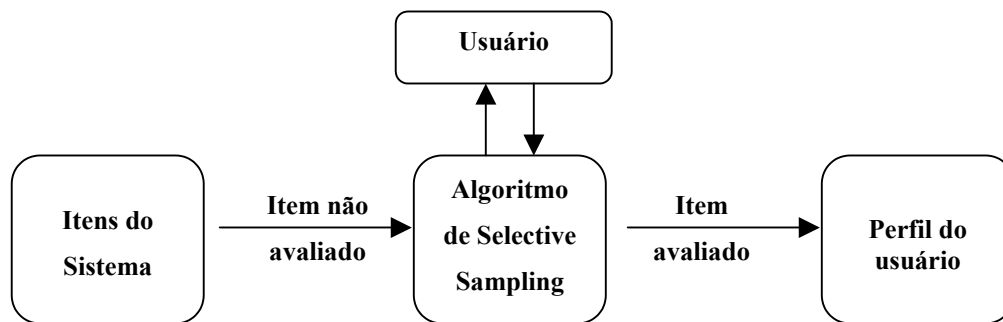


Figura 4.3 – Modelo de avaliação de um sistema de filtragem ativo. O algoritmo de *selective sampling* seleciona os itens que o usuário avalia.

O objetivo da aprendizagem ativa em filtragem de informação é encorajar o usuário a avaliar um conjunto reduzido de itens que possibilitem um aprendizado capaz de capturar o essencial sobre o seu perfil. Por exemplo, o sistema pode informar o usuário que um item é relevante para aprender sobre o seu perfil e pedir sua avaliação. Isso pode ser feito principalmente no momento que um novo usuário entra no sistema e nada é conhecido sobre ele. Através da avaliação de uma quantidade mínima de itens o sistema já teria conhecimento suficiente sobre o perfil do novo usuário para poder gerar recomendações pertinentes ao seu perfil.

O algoritmo de *selective sampling* depende do tipo de aprendizado que é feito. Nesta dissertação propomos uma metodologia de seleção das avaliações dos usuário para um sistema com filtragem colaborativa. Para isso primeiro deve ser apresentado o algoritmo de filtragem colaborativa utilizado. Uma análise do estado da arte de filtragem colaborativa e descrição do algoritmo adotado são descritos no capítulo 5.

4.6 Conclusão

Atualmente o perfil dos usuários é aprendido ou implicitamente através da observação do comportamento do usuário dentro do sistema, ou explicitamente através de avaliações dos itens do sistema. A avaliação implícita ainda não é bem explorada nos sistemas de recomendação, possivelmente por serem indicadores mais fracos da preferência. A avaliação explícita, apesar de ter uma aplicação mais ampla, acarreta um custo adicional ao usuário que se vê obrigado a avaliar uma quantidade grande de itens para obter um pequeno ganho na qualidade das recomendações.

A aprendizagem ativa foi introduzida como solução para problemas de aprendizagem de máquina nos quais a aquisição de exemplos para o aprendizado é dispendiosa, possivelmente por exigir esforço humano, como é o caso da aprendizagem do perfil em sistemas de recomendação através de avaliações explícitas.

Por último, neste capítulo, foi apresentado um modelo de avaliações de itens pelos usuários de um sistema de filtragem onde o sistema seleciona os itens que o usuário avalia para capturar o seu perfil de forma mais eficiente, diferente do modelo dos sistemas de filtragem atuais nos quais o sistema não tem qualquer controle sobre quais itens o usuário avalia.

Capítulo 5

Filtragem Colaborativa

Existem várias formas de fazer filtragem baseando-se na relação entre membros de uma comunidade que possuam perfil semelhante. Neste capítulo a filtragem colaborativa será discutida com ênfase na filtragem colaborativa com vizinhos mais próximos.

5.1 Definição do problema

A tarefa de um sistema de filtragem colaborativa é, para um dado usuário alvo e item, encontrar outros usuários que possuam afinidade com o usuário alvo no que diz respeito à informação filtrada e que tenham avaliado o item, com o objetivo de prever a avaliação do item pelo usuário alvo utilizando as avaliações já realizadas pelos outros usuário. Para tal tarefa o sistema deve coletar as avaliações dos usuários sobre as informações. As informações deverão estar organizadas em unidades de informação para as quais os usuários emitirão sua opinião e que receberão predições do sistema, formando os itens que serão filtrados. Por exemplo, em um sistema de filtragem de filmes, os itens são os filmes que recebem avaliações dos usuários e para os quais o sistema faz predições sobre a relevância para o usuário.

Os dados de um problema de filtragem colaborativa podem ser organizados em uma tabela de usuários (em linhas) versus itens (em coluna). Cada célula da tabela conterá o score da avaliação de um usuário para um dado item. As células da tabela que estão vazias representam os itens para os quais não é conhecida a avaliação do usuário. O objetivo da filtragem está em, quando requisitado, prever o valor de uma célula vazia da tabela utilizando os valores conhecidos da mesma. Um exemplo de matriz de avaliações está na figura 5.1.

	Item 1	Item 2	Item 3	Item 4
Usuário 1	2	4		5
Usuário 2		2	5	1
Usuário 3	3	1	5	?

Figura 5.1 – Exemplo da matriz de avaliações de usuários sobre itens em um intervalo de 1 a 5. O caractere “?” indica a requisição pela predição do item para o usuário.

Na tabela da figura 5.1 as células onde a avaliação do usuário para o respectivo item é conhecida estão preenchidas com o valor atribuído a este. No exemplo da figura 5.1 a avaliação do usuário 3 para o item 4 não é conhecida e se deseja obter uma predição para essa avaliação considerando o restante da tabela.

5.2 Algoritmos de Filtragem Colaborativa

O sistema Tapestry [Goldberg 92] é considerado por muitos o primeiro sistema com filtragem colaborativa. No Tapestry os usuários fornecem avaliações sobre os itens, no caso mensagens eletrônicas. Um usuário pode então consultar os itens baseando-se nas avaliações dos demais usuários. Nesse sistema o próprio usuário deve identificar quais são os usuários do sistema com os quais ele possui afinidade quanto à informação avaliada, e através das consultas formuladas sobre a avaliação desses usuário requisitar a informação. A desvantagem de tal sistema é que o usuário deve identificar outros usuários semelhantes, tornando-o prático apenas para grupos pequenos onde os usuários podem conhecer o perfil dos demais. Algum tempo depois surgiram os algoritmos para executar filtragem colaborativa automatizada, que são capazes de, a partir de uma tabela de avaliações, identificar a semelhança entre usuários e recomendar itens sem a necessidade de qualquer participação do usuário exceto aquela em que ele fornece suas avaliações.

Billus e Pazzani [Billsus 98] apresentaram um modelo de filtragem colaborativa com Redes Neurais. Para cada usuário do sistema é treinada uma rede que servirá como modelo para gerar futuras avaliações. Para construir a rede de um usuário são construídos exemplos de treinamento com os itens que ele avaliou. Em cada exemplo a entrada da rede é a avaliação dos demais usuários para o item e a saída a avaliação do usuário alvo. O problema dessa abordagem está no tamanho da entrada da rede que pode ser muito grande devido ao número de usuários no sistema. Billus e Pazzani tratam esse problema utilizando *singular value decomposition (SVD)* para reduzir a dimensão da tabela de avaliações.

Em [Lin 00] é utilizada a mineração de regras de associação para a filtragem. Para isso é utilizado um modelo de mineração de regras chamado de “classificação associativa”, que utiliza algoritmos de mineração de regras de associação para encontrar regras associativas que descrevem uma única classe. As regras extraídas para um usuário descrevem sua relação com os demais usuários e são utilizadas para fazer previsões. Por exemplo, as regras produzidas para um usuário qualquer podem ser do tipo “SE u_1 = gosta E u_2 = não gosta ENTÃO gosta” que significa que se o usuário u_1 gosta de um item e o usuário u_2 não gosta então o usuário alvo irá gostar. Cohen et al. [Cohen 98] também faz filtragem de filmes utilizando regras. Para extrair as regras sobre a relação de um usuário com os demais usuários do sistema é utilizado o algoritmo *Ripper* [Cohen 95] de aprendizagem indutiva de regras. As regras produzidas são somadas a regras que descrevem o conteúdo dos filmes que o usuário tem interesse para fazer uma filtragem híbrida.

Em [Breese 98] são usados algoritmos para aprender Redes Bayesianas. Cada nodo da rede é um item e o estado do nodo o valor possível de avaliação do item. Na rede aprendida cada item terá um conjunto de itens pais que são os melhores preditores para aquele item segundo as avaliações feitas pela comunidade de usuários. A previsão de avaliação de um usuário para um item receberá o valor (relevante ou não relevante) de maior probabilidade dada a tabela de probabilidade condicional extraída para o item da Rede Bayesiana aprendida e as avaliações do usuário.

Os algoritmos de filtragem colaborativa mencionados até agora envolvem a construção de um modelo de previsão. Eles são classificados como *Algoritmos Baseados em Modelo* [Breese 98]. Esses algoritmos ainda que muitas vezes eficientes na previsão necessitam construir um modelo antes de poder fazer previsões. Dependendo do algoritmo esse modelo pode ser uma rede neural, um conjunto de regras ou uma Rede Bayesiana. Entretanto a construção desses modelos envolvem dois problemas. Primeiro, o custo computacional de construir o modelo pode ser muito alto. Em um sistema de filtragem é comum que o número de usuários

seja na ordem de milhares. Na abordagem com redes neurais e com regras deve ser aprendido um modelo para cada usuário. Segundo, o surgimento de novas avaliações exige a construção de um novo modelo. A incorporação de novas avaliações aos modelos é uma possibilidade ainda não explorada nos algoritmos usados.

Uma alternativa para os algoritmos que constroem modelo são os *Algoritmos Baseados em Memória* [Breese 98]. Esses algoritmos, no momento da predição, operam sobre toda a base de usuários para fazer predições combinando as avaliações de usuários semelhantes. Ao contrário dos algoritmos baseados em modelos, os algoritmos de memória não necessitam da construção prévia de um modelo. Todo o processamento é realizado no momento em que uma predição é requisitada com a tabela de avaliações armazenada na memória. Estes algoritmos utilizam uma solução baseada nos algoritmos de KNN.

O sistema GroupLens [Resnick 94] de filtragem de mensagens de grupos de discussão foi o primeiro a prover esse tipo de filtragem. O sistema computa as similaridades do usuário alvo com os demais usuários. Os valores dessas similaridades são utilizados para selecionar uma amostra de usuários mais similares que são usados para predizer a avaliação de um item para o usuário alvo. A influência de cada usuário selecionado na predição dependerá da sua similaridade com o usuário alvo. Mais tarde outros sistemas foram desenvolvidos utilizando a mesma idéia, como os sistemas de recomendação de músicas Ringo [Shardanand 95] e de filmes Belcore Video Recommender [Hill 95]. Outros trabalhos mais recentes também envolvem esse tipo de filtragem [Cotter 00],[Herlocker 00] e [Billus 98]. O algoritmo baseado nos usuários mais próximos rapidamente apresentou grande aceitação devido a sua simplicidade e eficiência nas predições e ficou conhecido como *filtragem colaborativa com vizinhos mais próximos*. Atualmente esse algoritmo vem sendo utilizado em sistemas de recomendação com grande quantidade de usuários como o PTV [Cotter 00] ou o MovieLens [Good 99].

O trabalho apresentado nesta dissertação é voltado para esse tipo de algoritmo, que é apresentado em detalhes na seção seguinte.

5.3 Filtragem Colaborativa Com Vizinhos Mais Próximos

A predição de um item para um usuário alvo utilizando a filtragem colaborativa com vizinhos mais próximos pode ser decomposta em três etapas:

1. Calcular das similaridades entre o usuário alvo e os demais usuários do sistema.
2. Selecionar uma vizinhança formada pelos usuários mais similares ao usuário alvo que tenham avaliado o item em questão.
3. Realizar a predição do item para o usuário alvo, baseando-se na avaliação dos usuários que formam sua vizinhança.

A filtragem colaborativa com vizinhos mais próximos utiliza o método de KNN para realizar a predição de cada item para o usuário alvo. Para um dado item e usuário alvo, tem-se um KNN onde os exemplos são os usuários que avaliaram o item em questão, e a classificação de cada exemplo é a avaliação para o item feita pelo usuário que o exemplo representa. Para fazer a predição de um item, o sistema procura os k usuários mais próximos entre aqueles que avaliaram o item em questão. A figura 5.2 ilustra a situação.

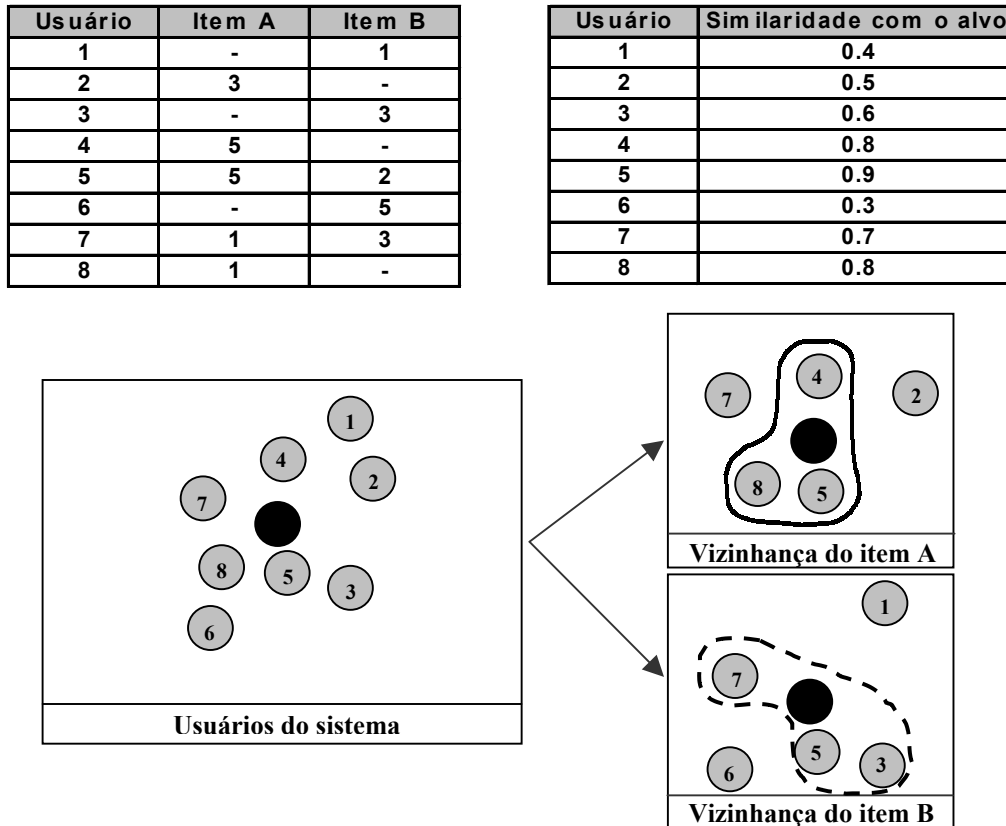


Figura 5.2 – Vizinhança utilizada na predição dos itens A e B para o usuário alvo (círculo preto). Para a predição de cada item, a vizinhança é retirada apenas dentre os usuários que avaliaram um item.

Os usuários do sistema são usados para prever a avaliação do usuário alvo (círculo preto) para os itens *A* e *B*. A tabela das avaliações mostra a avaliação de cada usuário do sistema para os dois itens. O traço indica que o usuário não avaliou o item. A tabela da similaridade indica a similaridade de cada usuário do sistema com o usuário alvo, que na figura é representada pela distância (mais próximo é mais similar). Para a predição de cada item, são usados apenas aqueles usuários que avaliaram o item em questão. Na figura 5.2 a predição dos itens *A* e *B* para o usuário alvo é feita com diferentes grupos de usuários do sistema. Para prever o item *A*, são usados apenas os usuários 2,4,5,7 e 8 que são aqueles que avaliaram esse item. A predição do item *A* é um KNN onde esses usuários são os exemplos e suas avaliações a classificação, e deseja-se prever a classificação (avaliação) do novo exemplo (usuário alvo). No exemplo é selecionada uma vizinhança de três usuário para a predição. Para o item *B*, a situação é a mesma, apenas com exemplos

diferentes, que no caso desse item serão os usuários 1,3,5,6 e 7. Sua vizinhança é selecionada dentre esses usuários.

Na filtragem colaborativa com vizinhos mais próximos o método KNN é executado para cada item que se quer fazer uma predição, a partir de diferentes exemplos para cada caso. Mesmo em casos onde o mesmo usuário é usado na predição de dois itens diferentes, ele pode representar exemplos diferentes em cada caso. O usuário 5, por exemplo, participa da predição dos itens *A* e *B*, entretanto em cada caso ele será um exemplo de classificação diferente porque possui avaliações diferentes para os dois itens.

Cada uma das etapas desse tipo de filtragem pode ser implementada de diferentes maneiras e estas terão efeito na predição. Aspectos sobre a implementação de cada uma das etapas são discutidos na seção seguinte.

5.4 Algoritmo de Filtragem Colaborativa

Para a implementação de um algoritmo de filtragem colaborativa com vizinhos mais próximos é necessário decidir como será realizada cada etapa da filtragem. As escolhas relativas a implementação das etapas baseiam-se em trabalhos existentes sobre esse tipo de filtragem.

A seguir são discutidas aspectos de implementação para cada uma das etapas da filtragem colaborativa com vizinhos mais próximos.

5.4.1 Similaridade entre os usuários

A similaridade entre os usuários é usada para medir o quanto dois usuários concordam quanto à relevância da informação filtrada. Essa similaridade é usada tanto para selecionar a vizinhança do usuário quanto para ponderar a influência desse usuário na predição do item. A similaridade entre dois usuários é calculada com base nos itens que eles avaliaram em comum, ou seja, com base na intersecção entre os conjuntos de itens avaliados por cada um.

A correlação de Pearson tem sido usada em vários sistemas de filtragem [Resnick 94],[Hill 95], [Billus 98], [Herlocker 00]. A correlação $s_{a,u}$ entre os usuários a e u é calculada para os m itens avaliados por ambos, onde $r_{a,i}$ e $r_{u,i}$ são as avaliações de a e u para um item i , e \bar{r}_a e \bar{r}_u são a média das avaliações de a e de u :

$$s_{a,u} = \frac{\sum_{i=1}^m [(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)]}{\sqrt{\sum_{i=1}^m (r_{a,i} - \bar{r}_a)^2 \sum_{i=1}^m (r_{u,i} - \bar{r}_u)^2}} \quad (\text{Eq. 5.1})$$

Estudos comparativos mostram que a correlação de Pearson apresenta os melhores resultados na filtragem colaborativa com vizinhos mais próximos [Herlocker 00][Breese 98]. Nesse trabalho ela é usada como medida de similaridade entre os usuários.

Um problema relacionado à similaridade entre os usuários em um sistema colaborativo está no número de avaliações que esses usuários possuem em comum. O cardinal da intersecção entre os conjuntos de avaliações de dois usuários pode variar de zero ao número de itens do sistema no caso onde dois usuários avaliam todos os itens. Quando o cardinal dessa intersecção é muito pequeno, a medida de similaridade pode julgar dois usuários muito similares com base em pouca informação. Por exemplo, se dois usuários possuem apenas dois itens avaliados em comum, e ambos avaliaram um item com escore 4 e outro com escore 5, então terão uma similaridade máxima, i.e. igual a 1. Entretanto se dois usuários possuem 50 avaliações em comum, das quais 49 eles concordam na sua avaliação e em uma eles discordam, Estes usuários terão similaridade inferior ao primeiro caso, i.e. menor que 1. Herlocker et al. [Herlocker 99] identifica a necessidade de considerar a quantidade de avaliações que dois usuários têm em comum no cálculo da similaridade, e sugere a utilização de uma penalização na função de similaridade de tal forma que quanto maior o número de avaliações em comum entre dois usuários menor será a penalização. Para tal, Herlocker et al. usam a seguinte forma de penalização para a similaridade entre dois usuários com m avaliações em comum:

$$s'_{a,u} = \begin{cases} \frac{s_{a,u}}{50 - m}, & \text{se } m < 50 \\ s_{a,u}, & \text{se } m \geq 50 \end{cases} \quad (\text{Eq. 5.2})$$

No algoritmo de filtragem que usamos, essa penalização é adotada por ter se mostrado melhor do que quando não é usada em testes preliminares, confirmando os resultados dos experimentos, usando a mesma base de dados, feitos por Herlocker [Herlocker 99], onde o número 50 usado na penalização foi encontrado experimentalmente.

5.4.2 Seleção dos usuários

Esta etapa envolve a escolha do conjunto de usuários que serão usados para a predição. No sistema GroupLens [Resnick 94], considerado o primeiro sistema de filtragem colaborativa com vizinhos, a predição de um item é feita utilizando todos os usuários que avaliaram o item em questão. No sistema Ringo, Shardanand et al. [Shardanand 95] advogam um limiar de similaridade para selecionar os usuários da vizinhança como uma melhor solução. Em trabalhos mais recentes Cotter et al. [Cotter 00] utilizam um número fixo de usuários na vizinhança, sendo selecionados aqueles de maior similaridade. Herlocker [Herlocker 00] analisa as formas de seleção de vizinhança concluindo que uma vizinhança de tamanho fixo apresenta melhores resultados na filtragem colaborativa, e sugere uma vizinhança de tamanho 40. Nesse trabalho decidimos utilizar uma vizinhança fixa de tamanho 40 por utilizarmos a mesma base de exemplos usada por Herlocker [Herlocker 99].

5.4.3 Geração de predições

Uma vez calculada a similaridade do usuário alvo com os demais usuários e selecionada a vizinhança com base nessa similaridade, resta combinar as avaliações

para gerar a predição. A maioria dos sistemas utiliza a similaridade com o usuário alvo para ponderar a contribuição de cada usuário [Konstan 97],[Billus 98]. Herlocker [Herlocker 00] faz um estudo comparativo entre algoritmos de predição ponderados ou não pela similaridade para concluir que a utilização da similaridade melhora a qualidade das predições. A função de predição, que é a mesma utilizada em [Konstan 97],[Billus 98], [Herlocker 00], é mostrada na equação 5.3:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n [(r_{u,i} - \bar{r}_u) w_{a,u}]}{\sum_{u=1}^n w_{a,u}} \quad (\text{Eq. 5.3})$$

A predição $p_{a,i}$ do item i para o usuário a será a média \bar{r}_a de suas avaliações dos itens da base mais um desvio. A direção e intensidade desse desvio é calculada, em uma vizinhança N de tamanho n , pela diferença entre a média \bar{r}_u das avaliações de cada usuário $u \in N$ em toda base de itens e sua avaliação $r_{u,i}$ para o item i , ponderada pela razão entre a similaridade $w_{a,u}$ com o usuário alvo e a soma das similaridades do usuário alvo em relação a cada usuário dessa vizinhança.

5.4.4 Algoritmo de Predição

Com base nas observações e nas escolhas feitas sobre as etapas da filtragem, introduzimos o algoritmo de filtragem colaborativa com vizinhos mais próximos usado que mostramos abaixo:

Entrada

a: usuário alvo
U: usuários do sistema excluindo o usuário alvo
i: item alvo

Saída

$p_{a,i}$: predição do item 'i' para o usuário 'a'

Predizer(a, i, U)

```

1.  Para cada usuário u em U
2.  begin
3.      Se  $i \in R_u$  então
4.      begin
5.           $Sim(a,u) \leq Pearson(a,u)$ 
6.          Se  $|R_a \cap R_u| < 50$  então  $Sim(a,u) \leq Sim(a,u) / (50 - |R_a \cap R_u|)$ 
7.          Lista  $\leftarrow (Sim(a,u), u)$ 
8.      end
9.  end
10. Ordenar os elementos de Lista em ordem decrescente segundo o valor de  $Sim(a,u)$ 
11. ListaVizinhos  $\leftarrow TopoLista(40, Lista)$ 
12.  $p_{a,i} \leftarrow Predicao( ListaVizinhos, i )$ 

```

Figura 5.3 – Algoritmo de filtragem colaborativa com vizinhos mais próximos.

No algoritmo da figura 5.3 a função $Pearson(a,u)$ retorna a correlação de Pearson calculada nas avaliações que os usuários possuem em comum conforme mostrado na equação 5.1. R_u e R_a indicam os conjuntos de itens avaliados pelo usuário u e pelo usuário alvo a respectivamente. O função $TopoLista(n, Lista)$ é uma função que retorna uma sublista da lista $Lista$ formada pelos n elementos que encabeçam a lista. A função $Predicao(Lista,i)$ utiliza os usuários e sua similaridade com o usuário alvo da lista $Lista$ para gerar uma predição de acordo com a equação 5.3.

No algoritmo da figura 5.3, no momento da predição para um dado usuário a e item i , a lista dos $n - 1$ usuários de U é percorrida na busca daqueles mais similares ao usuário a que possam formar sua vizinhança preditora para o item i . Para cada usuário $u \in U$ percorrido, a similaridade com o usuário alvo envolve encontrar a intersecção entre os conjuntos R_a e R_u dos itens avaliados por a e u . O cálculo da intersecção dependerá do tamanho do conjunto de itens avaliados pelos dois usuário e envolve uma busca linear de tamanho $m = \min(|R_a|, |R_u|)$. Com isso conclui-se que a complexidade do algoritmo para percorrer a lista de usuários e calcular a

similaridade de todos os usuários com o usuário alvo é de $O(mn)$. Depois de calculada a similaridade com todos os usuários, o sistema faz uma ordenação dos usuários para selecionar aqueles usuários mais próximos de a que serão utilizados para prever a avaliação de i . A ordenação com um algoritmo de *QuickSort* tem complexidade de ordem $O(n \log n)$ tomando uma média de casos de ordenações. O algoritmo final possui então uma complexidade de ordem $O(mn) + O(n \log n)$ que resulta em um algoritmo de complexidade de ordem $O(mn)$ se $\log n < m$.

Para a utilização do algoritmo da figura 5.3 em um sistema de recomendação, faz-se necessário adapta-lo para que possa gerar previsões para um conjunto de itens em vez de apenas um. Num sistema de recomendação normalmente existe um conjunto de itens para os quais são requisitadas previsões. O sistema utiliza as previsões para formar uma lista de recomendações. Para gerar previsões para um conjunto de itens, a lista de usuário é percorrida calculando a similaridade dos usuários do sistema com o usuário alvo. Os usuário da base são ordenados de acordo com sua similaridade com o usuário alvo. O custo para produzir essa ordem é de $O(mn)$ como mostrado anteriormente. Essa ordem é armazenada e utilizada na previsão de todos os itens, sendo portanto necessário calcular a similaridade com os usuários do sistema uma única vez para prever todos os itens.

Para cada item do conjunto de previsões requisitadas a lista ordenada é percorrida pela ordem de similaridade com o usuário alvo, do mais similar ao menos similar, até encontrar o número necessário para formar a vizinhança de usuários que avaliaram o item em questão. Para gerar a previsão de cada item, a lista de n usuários é percorrida em uma busca linear de ordem $O(n)$. Tendo-se um conjunto de k itens para fazer a previsão, então o algoritmo final terá complexidade da ordem de $O(kn) + O(mn)$ para prever todos os itens. Na maior parte dos sistemas entretanto, o número k de previsões requisitadas por recomendação é constante, como é o caso do sistema descrito no capítulo 7. Quando o tamanho de k for constante o algoritmo terá complexidade de ordem $O(mn)$.

5.5 Conclusão

Filtragem colaborativa é uma técnica de filtragem bastante eficiente onde estão ausentes diversas dificuldades próprias à filtragem baseada em conteúdo. Por exemplo, não existe a necessidade de descrição do conteúdo ou formas sofisticadas de avaliar e comparar o conteúdo de dois itens. É possível gerar boas previsões e diversificação apenas com a combinação da avaliação de usuários correlacionados.

Nesse capítulo foi mostrado o estado da arte de filtragem colaborativa. Também foi feita uma análise do algoritmo de filtragem com vizinhos mais próximos, que tem aplicação mais ampla em sistemas de filtragem colaborativa. Por último foi mostrado que o algoritmo adotado possui ordem de complexidade de $O(kn) + O(mn)$ para calcular a previsão de k itens em uma base de dados com n usuários que têm em média m avaliações em comum, ou em casos de sistemas de recomendação como o que será mostrado no capítulo 7 onde o tamanho de k é constante terá a ordem de complexidade de $O(mn)$.

Capítulo 6

*ActiveCP: Um método de aprendizagem ativa
para filtragem colaborativa*

Como já explicado, sistemas que utilizam filtragem colaborativa são capazes de prever a relevância de uma informação para um usuário, através da associação do seu perfil armazenado com o de outras pessoas de uma comunidade que compartilham preferências sobre um tipo de informação e possuem afinidade de preferências. O perfil de um usuário tem portanto papel fundamental na descoberta de pessoas dentro da comunidade que possuam afinidade de gostos e necessidades de informação para que possam servir de base para a geração de predições. Quanto mais informação sobre a preferência de um usuário, melhor será o conhecimento sobre o seu perfil e mais precisa será a identificação de usuários semelhantes, e conseqüentemente melhores serão as recomendações para ele. Entretanto, os sistemas atuais não possuem qualquer metodologia capaz de aprender de forma rápida e eficiente o perfil do usuário. Para se obter um pequeno ganho na qualidade das recomendação é necessário uma quantidade de avaliações cada vez maior. Neste capítulo é abordado o problema da necessidade de obtenção rápida e eficiente do perfil do usuário dentro de um sistema de recomendação. Para isso é proposta uma metodologia de aprendizagem do perfil através da seleção dos itens que um usuário avalia. A metodologia proposta é voltada para o algoritmo de filtragem colaborativa com vizinhos mais próximos.

6.1 *Selective sampling* para FC com vizinhos mais próximos

Para desenvolver um algoritmo de aprendizagem ativa para FC com vizinhos mais próximos, analisamos primeiro as metodologias usadas para selecionar exemplos em algoritmos de KNN.

6.1.1 Seleção de exemplos em KNN

O algoritmo KNN de aprendizagem baseado em instâncias assume que todas as instâncias de um problema correspondem a um ponto num espaço multidimensional. No momento de classificar um novo exemplo, o algoritmo utiliza uma função de distância para determinar quais exemplos cuja classificação é conhecida são mais

próximas desse exemplo, e usa a classificação dessa vizinhança para prever a classificação do novo exemplo. A capacidade de estimar o conceito para cada nova instância com KNN está relacionada a quais exemplos do domínio do problema o algoritmo conhece a classificação.

O problema de selecionar os exemplos que serão usados para estimar o conceito de cada nova instância em um KNN diz respeito a dois problemas relacionados a esse algoritmo de aprendizagem. Um deles é o problema de redução da base de exemplos. A redução em KNN é usada para diminuir o número de exemplos usados para a classificação sem comprometer a performance do classificador. Para isso um algoritmo de redução deve reter apenas os exemplos mais relevantes para futuras classificações e remover aqueles exemplos que não são essenciais para isso. A finalidade da redução é diminuir o espaço necessário para o armazenamento dos exemplos e tornar a classificação de novos exemplos mais rápida. Os algoritmos de redução utilizam o conceito de pontos de borda e pontos de centro para selecionar quais exemplos serão retidos [Wilson 00]. Pontos de centro são aqueles localizados dentro de um agrupamento de pontos com a mesma classificação dele. Pontos de borda são aqueles localizados em zonas de transição entre os agrupamentos de diferentes classificações. A estratégia dos algoritmos de redução consiste em selecionar dentre os pontos de centro e de borda aqueles que são mais importantes na classificação de futuros exemplos.

Outro problema relacionado ao algoritmo de KNN é o de selecionar exemplos para aprendizagem ativa. No trabalho de Hasenjager e Ritter [Hasenjager 98], é sugerida a seleção dos exemplos que estivessem no vértice do diagrama de Voronoi [Aurenhammer 91] construído com base nos exemplos classificados e na distância entre os mesmos. O diagrama de Voronoi para um conjunto de n pontos $X = \{x_1, \dots, x_n\}$ em um espaço D é a partição de D em n regiões $\{V(x_1), \dots, V(x_n)\}$ tal que $V(x_i) = \{x \in D : \forall x_j \in X, d(x, x_i) \leq d(x, x_j)\}$. Os vértices de um diagrama de Voronoi são os pontos cujos vizinhos mais próximos estão a uma mesma distância formando um círculo com o vértice no centro e os vizinhos na circunferência. A figura 6.1

ilustra o diagrama de Voronoi para um conjunto de pontos e um vértice no centro da circunferência com seus vizinhos mais próximos.

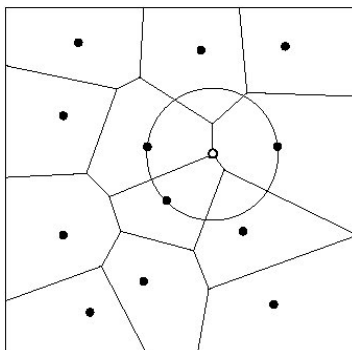


Figura 6.1 – Diagrama de Voronoi para um conjunto de pontos. Um vértice do diagrama como o centro do círculo formado por seus vizinhos mais próximos.

Hasenjager e Ritter [Hasenjager 98] investigaram três estratégias: selecionar os vértices aleatoriamente, selecionar os vértices mais distantes dos exemplos e selecionar os vértices mais distantes que tenham os vizinhos com classificação diferente. A última estratégia mostrou-se a mais eficiente por ser capaz de explorar bem o espaço de busca e selecionar os exemplos de fronteira. O trabalho de Hasenjager e Ritter é mais próximo do paradigma de *membership queries* [Angluin 88] pois os vértices de um diagrama são exemplos calculados pelo algoritmo e não selecionados de um conjunto de exemplos predefinidos. Entretanto, a estratégia pode ser adaptada para *selective sampling* selecionando os exemplos mais distantes dos seus vizinhos mais próximos cujos vizinhos tenham classificações diferentes [Lindenbaum 99].

O princípio explorado por essas metodologias de seleção está na relação de similaridade que um exemplo possui com os demais exemplos do sistema e na classificação de sua vizinhança. Nos algoritmos de redução, as noções de pontos de borda e pontos de centro usadas para determinar se um exemplo deve ser retido ou não são definidas pela similaridade desse exemplo com os demais exemplos e na classificação da sua vizinhança. Na aprendizagem ativa, a distância entre os exemplos é usada para determinar os vértices no diagrama de Voronoi e a

classificação dos exemplos vizinhos a um vértice usada para determinar a seleção desse exemplo.

6.1.2 Seleção de exemplos em FC com vizinhos mais próximos

Em um sistema de filtragem com vizinhos mais próximos, dado um usuário para o qual deseja-se melhorar a qualidade das predições, existem duas abordagens possíveis:

1. Adicionar novos usuários ao sistema que possam formar a vizinhança desse usuário.
2. Adicionar novas avaliações desse usuário, possibilitando que o sistema possa determinar com mais precisão quais usuários dentro do sistema são mais similares a ele.

A primeira alternativa tende a aumentar o número de usuários no sistema. Isso permite que, para cada item que se queira gerar uma predição, possivelmente haja uma quantidade maior de usuários que também avaliaram o item e que podem formar a vizinhança do usuário alvo para a predição desse item. Intuitivamente quanto maior o número de usuários em um sistema de filtragem colaborativa, melhor será a qualidade das recomendações. Em algoritmos de KNN o aumento do número de exemplos conhecidos tem efeito semelhante. Quanto mais exemplos classificados o algoritmo de KNN conhece, maior a chance de que haja exemplos no sistema similares a um novo exemplo para classifica-lo corretamente. O *selective sampling* em algoritmos de KNN é usado para selecionar os exemplos que são usados na classificação de forma a melhorar o desempenho do classificador. Esses mesmos métodos poderiam ser usados em filtragem colaborativa para selecionar novos exemplos.

Para um sistema de filtragem essa abordagem entretanto não é apropriada. A seleção de novos exemplos na filtragem colaborativa significa a seleção de novos usuários, que são os exemplos de avaliação de um item. Isso significa que para aprender sobre um item dever-se-ia selecionar os usuários e questioná-los sobre a

avaliação deste. Isso seria uma inversão de papéis, pois o objetivo do *selective sampling* na filtragem de informação é selecionar os itens que o usuário avalia para melhorar a recomendação fornecida a ele, e não selecionar os usuários que avaliam um item.

O segundo caso é mais apropriado ao que se propõe em um *selective sampling* para um sistema com FC. Dentre os itens existentes o sistema deve ser capaz de selecionar aqueles cuja avaliação ajudaria mais a capturar o perfil de um usuário em particular. Os algoritmos de seleção, como foi mostrado na seção 6.1.1, tem como requisito o conhecimento da distância entre os exemplos. Na seleção de itens em um algoritmo de FC a distância conhecida é entre os usuários do sistema, enquanto as avaliações aos itens são os atributos pelo qual é calculada a similaridade entre dois usuário. Para selecionar os itens devemos então criar um método de seleção que não dependa da distância entre esses itens.

Nesse trabalho desenvolvemos um método original de seleção dos itens que, quando avaliados pelo usuário, ajudam a aprender de forma mais eficiente sobre o seu perfil.

6.1.3 Critérios de seleção de itens em FC com vizinhos mais próximos

No algoritmo de FC com vizinhos mais próximos, quando o usuário a avalia um item qualquer, ele está em última instância modificando a forma com que ele está relacionado com os demais usuários do sistema que também avaliaram aquele item. Isso se dá porque a similaridade entre dois usuários é computada nos itens que eles avaliaram em comum. Aqueles usuários que avaliaram o item de forma igual, ou seja, com um escore igual ou próximo ao atribuído por a , estarão agora mais similares a ele, enquanto aqueles que avaliaram de forma diferente, ou seja, com um escore diferente do escore de a , estarão menos similares a ele.

Se considerarmos que cada usuário possui uma vizinhança ótima, i.e., o conjunto de usuários que são os melhores indicadores de suas futuras avaliações, seria necessário conhecer a avaliação de todos os itens por todos os usuário para

determinar essa vizinhança. Entretanto, como o usuário avalia apenas uma parcela dos itens do sistema, sua vizinhança deve ser determinada usando essas avaliações. Intuitivamente, em FC com vizinhos mais próximos, quanto maior o número de avaliações de um usuário melhor será determinada sua vizinhança, e conseqüentemente melhor será a qualidade das predições. Neste trabalho sugerimos que não só a quantidade de avaliações de um usuário alvo é importante para determinar seus vizinhos, mas também quais itens ele avalia. Definimos então o conceito de ganho de informação da avaliação de um item.

O ganho de informação da avaliação de um item para um usuário é definido como o quanto a avaliação desse item ajuda a determinar quais usuários do sistema são mais similares a esse usuário. Neste trabalho sugerimos duas medidas para selecionar itens cuja avaliação terá maior ganho de informação, ou seja, ajudará a determinar quais usuários são mais similares a um usuário e quais não são. São elas a controvérsia e a popularidade.

6.2 Controvérsia

O ganho de informação é usado em algoritmos de aprendizagem de máquina como na indução de árvores ID3 [Mitchel 97] para determinar qual atributo será usado na classificação em cada nível da árvore. Na indução de árvores, o ganho de informação mede o quanto um atributo ajuda a separar os exemplos de classificação através da seleção desse atributo. Para calcular o ganho de informação da classificação de um atributo é usada a entropia, que caracteriza a distribuição das classes em uma coleção de exemplos. Se a coleção de exemplos possui elementos com classificação variada, então diz-se com entropia maior do que se a classificação dos exemplos fosse concentrada em uma classe. A idéia do algoritmo de indução de árvores ID3 é selecionar os atributos que fornecem maior ganho de informação, ou seja, aqueles que diminuem a entropia devido à partição dos exemplos de acordo com os valores dos atributos.

Em algoritmos de aprendizagem ativa, a incerteza é utilizada na seleção de novos exemplos mais informativos [Lindenbaum 99]. Um exemplo é selecionado quando o grau de incerteza do classificador sobre sua classificação é alto. Essa estratégia permite que o erro de um classificador diminua na medida que são observados exemplos em regiões do domínio de exemplos onde o classificador não é preciso.

Em algoritmos de redução dos exemplos de um KNN, os pontos de borda se configuram como exemplos do domínio de exemplos localizados entre agrupamentos de classificações diferentes e a retenção desses pontos ajuda a determinar a fronteira entre dois agrupamentos [Wilson 00].

Em um sistema de filtragem, as avaliações realizadas sobre um item em um sistema se configuram como uma distribuição da opinião da população que utiliza o sistema sobre esse item. A forma com que as avaliações estão distribuídas vai determinar o quanto esse item é controvertido dentro da comunidade. Um item sobre o qual os usuários claramente não possuem unanimidade quanto à sua qualidade e que conseqüentemente recebe avaliações bastante diferenciadas é considerado controvertido dentro da comunidade. Neste trabalho a controvérsia é sugerida como característica análoga à incerteza usada em aprendizagem ativa, ou aos pontos de borda em algoritmos de redução de exemplos em KNN. Isso porque os itens controvertidos são aqueles cuja avaliação dos usuários não apresenta uma tendência clara, separando os usuários em grupos com diferentes opiniões sobre sua relevância. A controvérsia dos usuários na avaliação de um item é sugerida como critério para seleção de um item.

Em algoritmos de indução de árvores de decisão como o ID3, os atributos selecionados para a classificação são aqueles que maximizam o ganho de informação, ou seja, diminuem a incerteza sobre a classificação dos exemplos. Em um algoritmo de FC com vizinhos mais próximos um item controvertido, quando avaliado, fornece um ganho de informação sobre o perfil de um usuário, diminuindo a incerteza sobre a sua similaridade com os usuários da base.

A figura 6.2 mostra uma situação hipotética onde é mostrado o efeito de se avaliar um item controvertido:

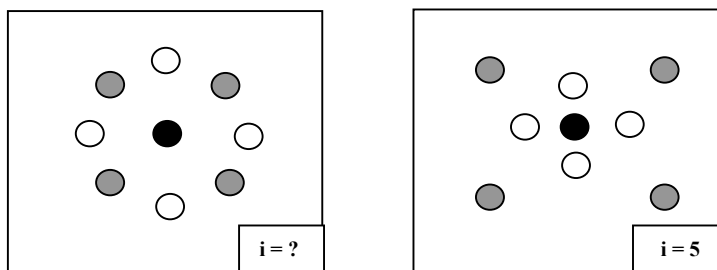


Figura 6.2– A relação do usuário alvo (preto) com os demais usuário antes e depois da avaliação do item i pelo usuário alvo. Os usuários brancos avaliaram o item i com escore 5 enquanto os usuários de cor cinza avaliaram o item i com o escore 1.

Na figura 6.2, a relação do usuário alvo (círculo preto) com os demais usuários é modificada pela avaliação do item i . Nesse exemplo quanto mais distante do usuário alvo, menor a similaridade em relação ao mesmo e vice-versa. Antes da avaliação desse item os usuários do sistema apresentavam a mesma similaridade em relação ao usuário alvo. Após a avaliação do item i pelo usuário alvo com o escore 5, os usuários que avaliaram o item i também com o escore 5 (círculos brancos) se tornaram mais similares ao usuário alvo, enquanto os usuários que o avaliaram com o escore 1 (círculos cinzas) se tornaram mais dissimilares em relação ao usuário alvo. A avaliação do item i portanto contribui para diminuir a incerteza sobre quais usuários possuem perfil similar ao usuário alvo. O fato de o item i ser controvertido, ou seja, dos usuários se dividirem quanto à sua avaliação (uma parte avaliou com 5 e outra com 1), permitiu que a avaliação desse item pelo usuário alvo ajudasse a ganhar informação sobre quais usuários são mais similares ao usuário alvo. A figura 6.3 mostra a situação de um item em que todos os usuário avaliaram com a mesma nota é avaliado pelo usuário alvo:

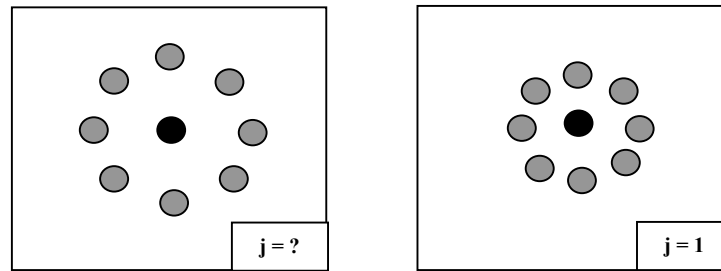


Figura 6.3 - A relação do usuário alvo (preto) com os demais usuário antes e depois da avaliação do item j pelo usuário alvo. Todos os usuários de cor cinza avaliaram o item j com o escore 1, o que indica o consenso na avaliação desse item.

No exemplo da figura 6.3, quando o usuário avalia o item j com o escore 1, a determinação de quais usuários são mais similares a ele continua a mesma de antes dele realizar essa avaliação, ou seja, o ganho de informação é nulo. Isso acontece porque a opinião dos usuários do sistema sobre o item j estão concentradas, os usuário têm o consenso em torno de uma avaliação com escore 1 para esse item. Nesse caso, mesmo que o usuário alvo avaliasse os itens com um escore diferente, como por exemplo, com o escore 5, a similaridade entre ele e os usuários da base iria diminuir, mas a informação sobre quais usuários são mais similares a ele e quais não são permaneceria a mesma.

A utilização da controvérsia como critério de seleção dos itens que um usuário avalia levanta algumas questões práticas que discutiremos a seguir.

6.2.1 O que é controvérsia ?

A primeira questão sobre a controvérsia como critério de seleção é quando um item é considerado controvertido. Para cada item existe uma distribuição das avaliações dos usuários sobre sua relevância. A figura 6.4 mostra quatro possíveis distribuições de avaliações em um intervalo de 1 a 5:

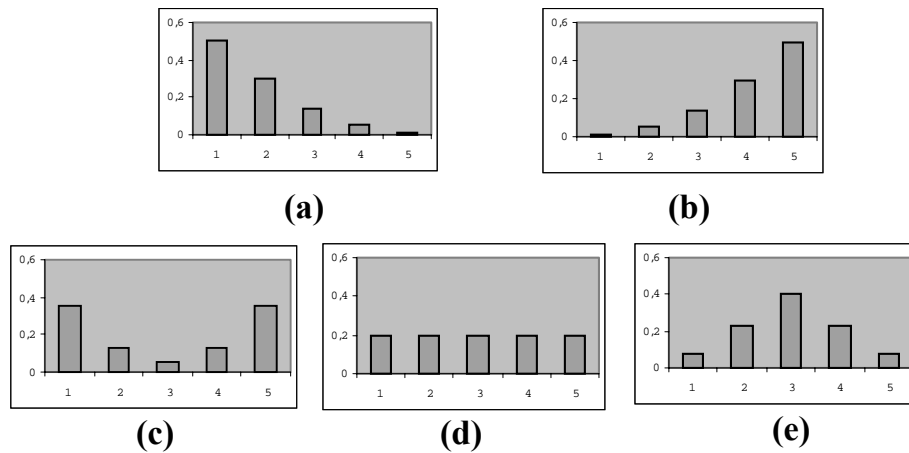


Figura 6.4 - Exemplos de distribuições das avaliações para itens com escores entre 1 a 5. Valores maiores indicam maior relevância ou preferência dos usuário para um item.

As distribuições mostradas em (a) e (b) apresentam claramente uma tendência dos usuários quanto à avaliação de um item como relevantes ou não relevantes. Itens com essa distribuição claramente não podem ser considerados controvertidos entre os usuários. Já nas distribuições de avaliações mostradas em (c) e (d) não há uma indicação clara de qual a preferência da maioria. Na situação (c), a maior parte dos usuários avalia o item com o escore mínimo (1) ou com o escore máximo (5), o que claramente indica que este item é controvertido. Na situação (d) as avaliações dos usuários se distribuem igualmente em todo intervalo de avaliação. Nesse caso também pode-se dizer que não há uma concordância entre os usuários sobre a relevância desse item, ou seja, o item é controvertido. A questão que surge então é em qual das duas situações pode-se dizer que o item é mais controvertido, ou seja, se temos dois itens, um com a distribuição das avaliações igual a (c) e outro igual a (d), qual deles devemos selecionar? Uma última situação seria a que é mostrada em (e). Nesse caso os usuário tendem a avaliar um item com uma avaliação intermediária. Esse comportamento dos usuários indica uma incerteza sobre a relevância de um item, o que pode levar a interpretá-lo como controvertido.

A questão do que é a controvérsia conduz naturalmente ao problema de como medir a controvérsia, que será visto na próxima seção.

6.2.2 Como medir a controvérsia ?

A partir do que se considera controvertido deve-se escolher uma métrica que traduza essa observação. Existem diversas medidas de dispersão que podem ser utilizadas para medir como as avaliações dos usuários se distribuem. O conceito do que é considerado controvertido ou não dependerá da métrica usada.

Neste trabalho utilizamos a variância como métrica para calcular a controvérsia das avaliações de um item. A controvérsia c_i do item i é calculada da seguinte forma:

$$c_i = \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_i)^2}{n} \quad (\text{Eq. 6.1})$$

Para um conjunto de n usuários, $r_{u,i}$ é a avaliação do item i pelo usuário u , e \bar{r}_i é a média das avaliações feitas para o item i . A variância como medida de controvérsia mede o quando está dispersa a distribuição das avaliações em dado conjunto de avaliações de usuários.

A figura 6.5 mostra a variância das distribuições da figura 6.4:

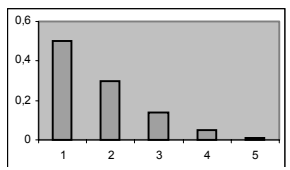
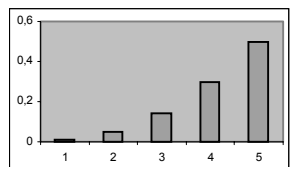
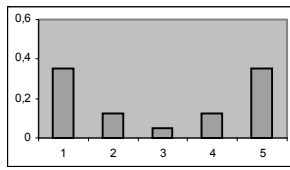
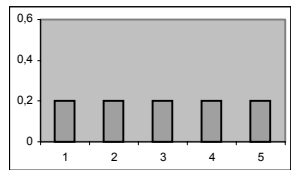
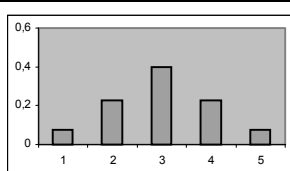
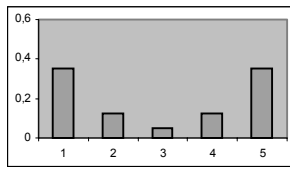
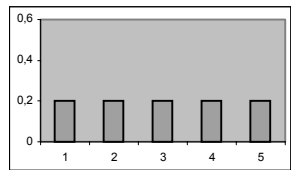
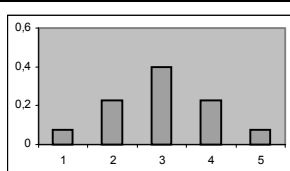
Distribuição		Variância	Distribuição		Variância
(a)		0,87	(b)		0,87
					
					
(c)		3,05	(d)		2,00
(e)		0,90			

Figura 6.5 – Variância para as distribuições mostradas na figura 6.4

As distribuições (a), (b) e (e) possuem uma variância pequena indicando que os itens com essa distribuição de avaliações não são controvertidos. As distribuições (c) e (d) possuem maior variância e são consideradas controvertidas. A distribuição (c) tem variância maior do que a de (d), o que indica que a escolha da variância como métrica da controvérsia fará com que os itens cujas distribuições de notas segundo os usuários expressem mais acentuadamente grupos de opiniões contrárias sejam considerados mais controvertidos.

Uma possível modificação na variância como medida da controvérsia é utilizar a similaridade com os usuários no seu cálculo. A equação 6.2 apresenta uma outra métrica derivada da variância cujos termos são ponderados pela similaridade dos usuários com o usuário alvo:

$$c_{a,i} = \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_i)^2 w_{a,u}}{n \sum_{u=1}^n w_{a,u}} \quad (\text{Eq. 6.2})$$

A controvérsia de um item i para o usuário alvo a será ponderada pela similaridade $w_{a,u}$ de a com o usuário u . Essa modificação da variância faz com que os usuários mais similares ao usuário alvo tenham maior peso quando calculada a controvérsia de um item.

Existem outras medidas de dispersão que podem ser usadas para medir a controvérsia de um item. Neste trabalho escolhemos usar a variância da equação 6.1 como medida da controvérsia por traduzir o que intuitivamente consideramos controvertido na distribuição das avaliações de um item. A utilização da equação 6.2 requer o conhecimento prévio do perfil do usuário para calcular os pesos $w_{a,u}$, o que impossibilita de utilizá-la quando o usuário não avaliou item algum. A utilização dessa equação é citada como trabalho futuro.

6.2.3 Controvérsia Global versus Controvérsia Local

A distribuição das avaliações feitas por um conjunto de usuários a um item determina a sua controvérsia. Para construir essa distribuição entretanto, é necessário verificar qual é o conjunto de usuários cuja avaliações serão consideradas.

A primeira idéia quando se fala em medir a controvérsia de um item no sistema é utilizar a avaliação de todos os usuários do sistema que avaliaram o item, ou seja, a controvérsia global. Dessa forma um item será considerado controvertido quando a distribuição das avaliações de todos os usuários for dispersa.

Uma alternativa à controvérsia global é a controvérsia local. A controvérsia local refere-se à distribuição das avaliações a um item apenas entre os usuários mais similares ao usuário alvo. Nesse caso a distribuição de um item implicará que ele seja controvertido ou não para um usuário dependendo de quais usuários são mais similares a ele. O tamanho da vizinhança de um usuário usada para medir a controvérsia deve ser menor do que o número de usuários do sistema, pois caso contrário a controvérsia seria global. Para calcular a controvérsia local sugerimos usar uma vizinhança igual ou um pouco maior do que a vizinhança usada nas predições ¹.

A justificativa intuitiva para usar a controvérsia local é que um item pode ser consensual na comunidade mas controvertido entre os vizinhos de um usuário, ou o contrário também pode ocorrer. Como as predições de um item são feitas baseada nos usuários mais similares, o ganho de informação sobre a similaridade dos vizinhos de um usuário será mais relevante, o que leva a crer que a distribuição das avaliações entre os usuários mais similares terá mais importância na seleção dos itens que um usuário avalia.

¹ Neste trabalho usamos uma vizinhança preditora de tamanho 40 como mostrado no capítulo 4. A utilização de uma vizinhança desse tamanho é porque a idéia da controvérsia local é calcular a controvérsia com os mesmos usuários que serão usados nas predições.

6.2.4 Cobertura versus Intensidade da Controvérsia

Em um sistema de filtragem, o número de avaliações para os diferentes itens pode variar bastante. Um item pode ser avaliado por apenas dois usuários que tem opiniões radicalmente opostas sobre sua relevância. Nesse caso esse item seria considerado muito controvertido, o que não corresponde à noção de controvérsia proposta. O que as medidas de controvérsia como a variância medem é apenas a intensidade da controvérsia. Outro aspecto que deve ser considerado na controvérsia é a sua cobertura, ou seja, quantos usuários participam de seu cálculo.

Para considerar as noções de intensidade e cobertura no cálculo da controvérsia propomos fixar o número de usuários usados para seu cálculo. Fixando o número de usuários utilizados garantimos que a cobertura no cálculo da controvérsia de todos os itens será igual, tornando a medida da controvérsia focada na sua intensidade. Para selecionar uma quantidade fixa de k de usuários que avaliaram um item para calcular sua controvérsia propomos utilizar estratégias diferentes quando a controvérsia é local ou global. Na controvérsia local o número de usuários requeridos é retirado da sua vizinhança, ou seja, para usar k usuários para calcular a controvérsia de um item i são selecionados os k usuários mais similares ao usuário alvo que avaliaram o item i . No caso da controvérsia global, os usuários usados para calcular a controvérsia de um item são uma amostragem dos usuários do sistema que avaliaram esse item, ou seja, para usar k usuários para calcular a controvérsia de um item i são selecionados aleatoriamente k usuários dentre aqueles que avaliaram o item i .

As respostas para questões sobre a controvérsia apresentadas aqui bem como sobre a eficácia da mesma como critério de seleção devem ser obtidas experimentalmente. Antes disso analisaremos um segundo critério de seleção dos itens que um usuário avalia, a popularidade.

6.3 Popularidade

Na FC com vizinhos mais próximos, a similaridade entre dois usuários é calculada nos itens que eles avaliaram em comum. A credibilidade que se tem na similaridade calculada entre dois usuários deve ser proporcional ao cardinal da intersecção dos conjuntos de avaliações dos dois, como mostrado no capítulo 5. Quando um par de usuários tem um número grande de avaliações em comum, o sistema deve ter mais credibilidade nessa similaridade do que quando possuem um número pequeno de avaliações em comum. Neste trabalho a credibilidade da similaridade é tratada no cálculo da similaridade entre dois usuários através da penalização feita quando dois usuários têm poucas avaliações em comum. Essa penalização mostrou-se mais efetiva em um algoritmo de FC com vizinhos mais próximos do que o mesmo algoritmo sem a penalização [Herlocker 00].

Tendo em vista a importância que o número de avaliações em comum apresentadas por dois usuários tem na cálculo da similaridade entre eles, a popularidade de um item que um usuário avalia passa a ser relevante para determinar a vizinhança desse usuário.

O número de avaliações realizadas pelos usuários sobre um item é o que define a sua popularidade¹ no sistema. Nesse trabalho sugerimos que a popularidade de um item é um critério relevante na seleção dos itens que um usuário avalia. A idéia em selecionar os itens populares é que, quando um usuário avalia um item que outro usuário avaliou, há um ganho de informação sobre a similaridade desses dois usuários. Portanto, quanto maior o número de usuários que avaliou um item, maior será o ganho de informação sobre a similaridade do usuário alvo com relação aos usuários da base. A avaliação de um item que poucos usuários avaliaram fornece informação sobre similaridade do usuário alvo com poucos usuários da base, sendo assim menor o ganho de informação.

¹ A popularidade de um item aqui refere-se à quantidade de usuários que avaliou um item. A popularidade de um item não está relacionada ao conceito dos usuários da comunidade sobre a qualidade de um item como poderia se imaginar, mas apenas ao número de usuários que avaliaram um item.

O critério da popularidade é mais simples que a controvérsia. Das questões levantada sobre a utilização da controvérsia na seleção dos itens, apenas a questão de local e global se aplica ao critério da popularidade.

A popularidade de um item depende de quais usuários do sistema são considerados relevantes para determinar essa popularidade. Uma possibilidade é considerar a popularidade de um item em todos os usuários do sistema, ou seja, para ser considerado popular um item deve ser avaliado com frequência pelos usuários do sistema.

Uma alternativa à popularidade global é popularidade local. A popularidade local refere-se apenas aos usuários mais similares a um usuário alvo, ou seja, para um item ser considerado popular ele deverá ser avaliado com frequência pelos usuários mais similares a ele. Um item pode ser popular na comunidade mas não entre os vizinhos de um usuário, ou o contrário também pode ocorrer. Como as predições de um item são feitas baseada nos usuários mais similares, o ganho de informação sobre a similaridade dos vizinhos de um usuário será mais relevante, o que leva a crer que a popularidade entre os usuários mais similares terá mais importância na seleção das avaliações de um usuário.

6.4 Análise Empírica

Após a reflexão feita sobre a seleção dos itens em um algoritmo de *selective sampling* para FC com vizinhos mais próximos, nos propomos a avaliar os critérios de controvérsia e popularidade propostos. Nessa seção são descritos dados, metodologia e testes realizados.

A forma mais apropriada de testar uma metodologia de seleção seria com a iteração de usuários reais que avaliariam os itens indicados pelo sistema. A principal vantagem desse teste é que simularia uma situação real onde existem fatores como a incapacidade do usuário de avaliar itens que ele desconhece. Entretanto, isso exigiria um trabalho exaustivo de repetições de testes de seleções por parte dos usuários, além de ter limitações quanto ao número de usuários testados e à

quantidade de itens que um usuário avalia. Uma alternativa seria obter as avaliações dos usuários para todos os itens do sistema possibilitando executar repetidos testes de seleção sem a participação do usuário. O problema dessa metodologia está em obter avaliações para todos os itens, pois o número de itens em um sistema pode ser muito grande. Para o escopo desse trabalho tomamos uma base pública de um sistema de recomendação para executar os testes de seleção e criamos uma metodologia de teste baseada nesses dados.

6.4.1 Base de Dados

A base de dados utilizada [McJones 97] é formada por avaliação de usuários reais sobre filmes. A base é formada por 72.916 usuários e 1.628 filmes. Cada usuário avaliou no mínimo 20 filmes, em uma escala de 1 a 5, onde os valores correspondem às avaliações péssimo (1), ruim (2), regular (3), bom (4) e ótimo (5). Para os testes realizados neste trabalho utilizamos uma parte dos usuários da base formada por 10.000 usuários selecionados aleatoriamente. O número menor de usuários foi usado por representar uma diminuição no tempo dos experimentos e no espaço ocupada pela base usada sem maiores prejuízos aos resultados, uma vez que os recursos disponíveis para os experimentos eram limitados.

6.4.2 Métricas

As métricas usadas foram ROC e Breese apresentadas no capítulo 3. Para a Breese foi utilizado um *half-life* de 5. O valor de d é igual a 3, escolhido por ser o valor médio no intervalo de avaliação sendo assim aquele que mais aproxima a neutralidade de preferência do usuário. Na métrica ROC foram consideradas as avaliações com valores 1, 2 e 3 como insatisfatórias e avaliações com valores de 4 e 5 como satisfatórias.

6.4.3 Organização dos experimentos

Para testar a eficiência da seleção das avaliações partimos de uma visão simplificada do problema aprendizagem ativa em um sistema de recomendação. Na nossa metodologia o algoritmo deve selecionar os itens apenas entre aqueles em que já são conhecidas as avaliações do usuário testado. Dessa forma o problema do usuário não conhecer um item selecionado é eliminado.

Dentre os usuários da base de dados usada, foram selecionados aleatoriamente 1000 usuários que tinham efetuado avaliações em pelo menos 100 itens. Para cada um desses usuários, foram selecionadas aleatoriamente 100 dos itens avaliados pelo usuário em questão. Para medir o desempenho de uma metodologia de seleção dos itens que o usuário avalia, a mesma é empregada no estudo de cada um dos 1000 usuários selecionados.

Os itens dos usuários são divididos em 5 conjuntos de 20 itens cada. Os cinco conjuntos são usados para executar uma validação cruzada com 5 iterações. Em cada uma das execuções, um dos cinco conjuntos de avaliações do usuário é designado como conjunto de teste, conjunto de avaliações independente usado para medir o desempenho do algoritmo de seleção através de uma das métricas escolhidas (Breese ou ROC). Os outros quatro formam o conjunto de seleção. Em cada execução dispõe-se de um total de 80 avaliações no conjunto de seleção. O método de seleção testado deve então, para cada execução, selecionar uma parte dos itens do conjunto de seleção que considera mais relevantes para determinar o perfil do usuário. As avaliações dos itens selecionados são utilizadas para predizer as avaliações do conjunto de teste utilizando o algoritmo de FC com vizinhos mais próximos descrito no capítulo 5. O algoritmo mostrado na figura 6.6 mostra a execução do teste para cada usuário selecionado.

Entrada

U[1..5]: conjuntos originais de itens avaliados pelo usuário
n: número de itens a selecionar

Saída

a: precisão das predições

TesteSelecaoUsuario(U[1..5] , n)

```

1.   Para i = 1 to 5
2.   begin
3.       TestSet T<- U[i] //um conjunto de U
4.       SelectionSet S <- os outros 4 conjuntos de U
5.       EvaluationSet E <-∅.
6.       Enquanto |E| < n
7.       begin
8.           E <- SelecionarItem(S,E)
9.       end
10.      P <- Predizer(T,E)
11.      a[i]<- CalcularPrecisão(P,T)
12.   end
13.   Retornar a precisão média de a[i], i = 1..5

```

Figura 6.6 – Algoritmo do teste de seleção dos itens que um usuário avalia

O método *TesetSelecaoUsuario()* recebe como parâmetros um vetor $U[1..5]$ com os cinco conjuntos de itens que o usuário avaliou e o número n de itens que serão selecionados a cada iteração da validação cruzada. O método executa 5 iterações da validação cruzada. Para cada iteração um dos conjuntos de $U[1..5]$ é selecionado como conjunto de teste T e os 4 restantes juntos formam o conjunto de seleção S . O método de seleção *SelecionarItem()* seleciona um item de S que não está em E , e insere no conjunto E que inicialmente está vazio. A seleção de itens se repete até atingir o número n de itens inseridos em E . A avaliação dos itens inserido no conjunto E são então usadas pelo algoritmo de FC com vizinhos mais próximos para gerar predições P das as avaliações dos itens do conjunto T . A precisão das predições dos itens do conjunto T é calculada para cada uma das 5 iterações da validação cruzada. O algoritmo enfim retorna a precisão média calculada nas 5 iterações.

A figura 6.7 ilustra o modelo geral usado para os experimentos:

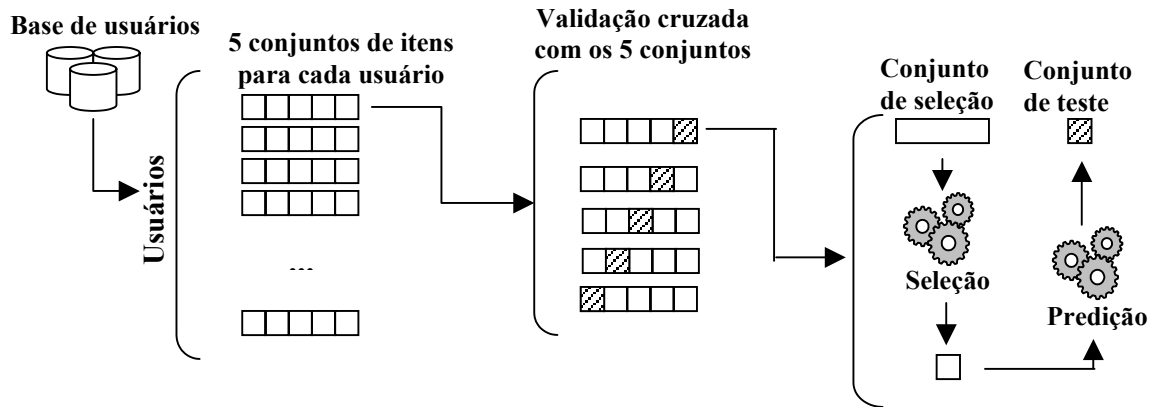


Figura 6.7 – Metodologia de teste dos algoritmos de seleção

São selecionados inicialmente os 1000 usuários da base de usuários. Os 100 itens avaliados pelos usuários são divididos em 5 conjuntos. Para cada usuário o algoritmo da figura 6.6 é executado realizando a validação cruzada com os 5 conjuntos do usuário e retornando a precisão média das predições usando as avaliações dos itens selecionados. O desempenho final de uma metodologia será a média do valor retornado pelo algoritmo 6.6 para os 1000 usuários testados.

6.4.4 Teste de hipótese

A eficiência de uma metodologia de seleção de itens é calculada pelo desempenho do algoritmo de predição utilizando as avaliações dos itens selecionadas no conjunto de seleção para prever as avaliações do usuário. Os valores médios de cada uma das métricas ROC e Breese para cada um dos 1000 usuários pode ser vista como uma realização de uma variável aleatória de distribuição aproximadamente normal. Dispõe-se então de uma amostra de tamanho 1000 dessa variável aleatória. O desempenho global para uma dada metodologia de seleção é expresso como a média dos valores médios de cada uma das métricas ROC e Breese calculados para cada um dos 1000 usuários e é também uma variável aleatória de distribuição normal segundo o teorema central do limite em estatística. Para verificar se existe evidência estatística em relação a hipótese de superioridade de uma metodologia em relação a outra utilizamos um teste de hipótese.

Uma hipótese estatística é uma afirmação a respeito de um parâmetro de uma distribuição (população). Um teste de hipótese é um procedimento para, a partir de uma amostra aleatória e do cálculo de um teste estatístico apropriado, decidir se uma hipótese estatística deve ou não ser rejeitada. No nosso caso, como o desempenho de uma metodologia de seleção de avaliações é expressa como a média do desempenho dessa metodologia aplicada a cada usuário (valor médio das métricas ROC ou Breese), sendo uma variável aleatória de distribuição normal, o teste estatístico apropriado é um teste unilateral de diferença de médias. A hipótese nula é a igualdade das médias (metodologias de seleção de avaliações de mesmo desempenho) e a hipótese alternativa é a superioridade de uma das médias (superioridade de uma das metodologias em relação a outra). A variável do teste normalmente seria a t de Student, mas como o tamanho da amostra é muito grande (1000 usuários), pode-se usar diretamente a normal padrão.

Para mostrar o resultado do teste de hipótese foi utilizado o *P-value*. O *P-value* é a probabilidade de que o teste estatístico vai tomar um valor que seja tão extremo quanto o valor observado quando a hipótese é nula. O *P-value* é o menor nível de significância α que levaria a rejeitar a hipótese nula, ou seja, o valor p tal que a hipótese nula seja rejeitada se $\alpha \geq p$ [Montgomery 97].

Os resultados de um teste de hipótese sobre a performance de duas metodologias de seleção fornecerão, para cada métrica utilizada, o seu valor médio x , o desvio padrão s encontrado no experimentos para os 1000 usuários testados, e o *P-value* p para um teste com a hipótese nula de igualdade entre as médias. Neste trabalho, a hipótese nula será rejeitada quando o valor de p for menor ou igual a 0,1, ou seja, a um nível de significância de 10%.

6.4.5 Performance

Nessa seção são mostrados os resultados dos experimentos com os algoritmos de seleção dos itens avaliados pelo usuário.

Seleção Aleatória

A seleção aleatória das avaliações é usada como referência na comparação das metodologias de seleção. O objetivo de um método de seleção é prever as avaliações desconhecidas com o algoritmo de FC com vizinhos mais próximos utilizando as avaliações dos itens escolhidas por ele, de forma mais eficiente do que se esses itens tivessem sido selecionadas aleatoriamente.

Neste trabalho a seleção aleatória foi selecionada como referência por ser a forma mais próxima do modelo de avaliação que os sistemas de recomendação utilizam atualmente, ou seja, os usuários selecionam quais itens serão avaliados de forma não determinística. De fato, a seleção aleatória aproxima o modelo de avaliação em um sistema de filtragem passivo mostrado na figura 4.1. Nesse modelo, os itens avaliados pelo usuário são determinados por diversos fatores sobre os quais o sistema não tem controle. O usuário avalia itens que surgem no sistema ao longo de sua utilização. A forma com que o usuário irá navegar pelo sistema, quais itens serão mostrados para ele e quais serão avaliados não podem ser determinados.

Seleção baseada na controvérsia

Para os experimentos mostrados foi utilizada como métrica da controvérsia a variância das avaliações para um item. Por ser uma abordagem mais simples, foi assumida uma controvérsia global, ou seja, calculada entre os usuários do sistema sem observar sua similaridade com o usuário alvo. Para tratar da questão da cobertura e intensidade no cálculo da controvérsia de um item, tomamos diferentes tamanhos de amostras de usuários do sistema que avaliaram esse item. Os tamanhos de amostra usados foram de 100 e 1000 usuários, além da controvérsia total que refere-se à utilização das avaliações de todos os usuários que avaliaram um item para calcular sua controvérsia. A figura 6.8 mostra o gráfico da performance das predições quando as avaliações são selecionadas segundo a controvérsia dos itens calculada em cada um dos três casos. A tabela 6.1 resume os dados desse gráfico.

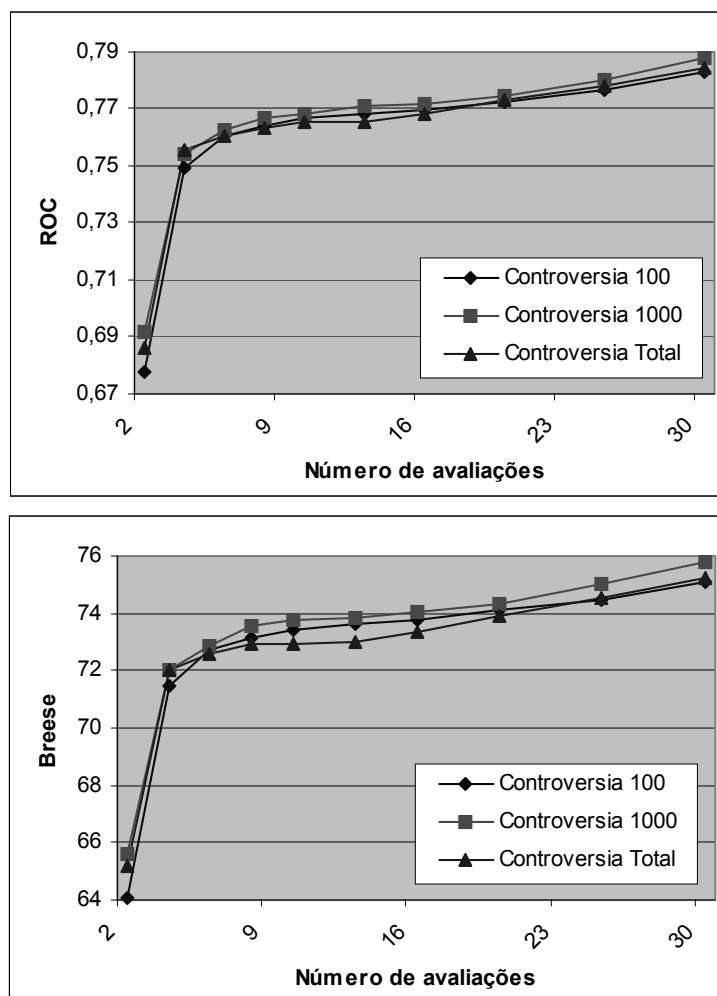


Figura 6.8 – Gráfico da performance das predições usando as avaliações dos itens selecionados pelo critério da controvérsia calculada em amostras de 100 e 1000 usuário e em todos os usuários. Para uma quantidade de itens selecionados (eixo x), mostra-se o desempenho do algoritmo no conjunto de teste (eixo y).

Número de avaliações	ROC			Breese		
	Controvérsia 100	Controvérsia 1000	Controvérsia Total	Controvérsia 100	Controvérsia 1000	Controvérsia Total
2	0,67750	0,69195	0,68622	64,097	65,583	65,155
4	0,74914	0,75440	0,75541	71,462	72,049	72,035
6	0,76071	0,76250	0,76050	72,697	72,828	72,606
8	0,76428	0,76685	0,76365	73,120	73,539	72,912
10	0,76656	0,76855	0,76510	73,434	73,739	72,962
13	0,76838	0,77081	0,76546	73,653	73,863	73,013
16	0,76992	0,77156	0,76799	73,791	74,0492	73,355
20	0,77214	0,77484	0,77305	74,142	74,3319	73,912
25	0,77644	0,78002	0,77808	74,435	74,9959	74,521
30	0,78268	0,78766	0,78428	75,115	75,7686	75,233

Tabela 6.1 – Tabela que mostra os valores médios de ROC e Breese para as metodologia de seleção com controvérsia em uma amostra de 100 e 1000 usuários e com todos os usuários do sistema.

Das metodologias de seleção baseada na controvérsia, a que apresentou melhor resultado em média foi a que utiliza a controvérsia calculada em uma amostra de 1000 usuários. A diferença dos valores não é significativa o suficiente para comprovar estatisticamente a superioridade dessa metodologia em relação as outras. Porém existe uma evidência de que a utilização dessa metodologia é mais eficiente pois a sua performance média está quase sempre acima das demais metodologias. A controvérsia calculada em uma amostra de 1000 usuários é escolhida para calcular a controvérsia.

Para comparar a metodologia de seleção escolhida com a metodologia de seleção aleatória os gráficos do desempenho das duas metodologias são mostrados na figura 6.9:

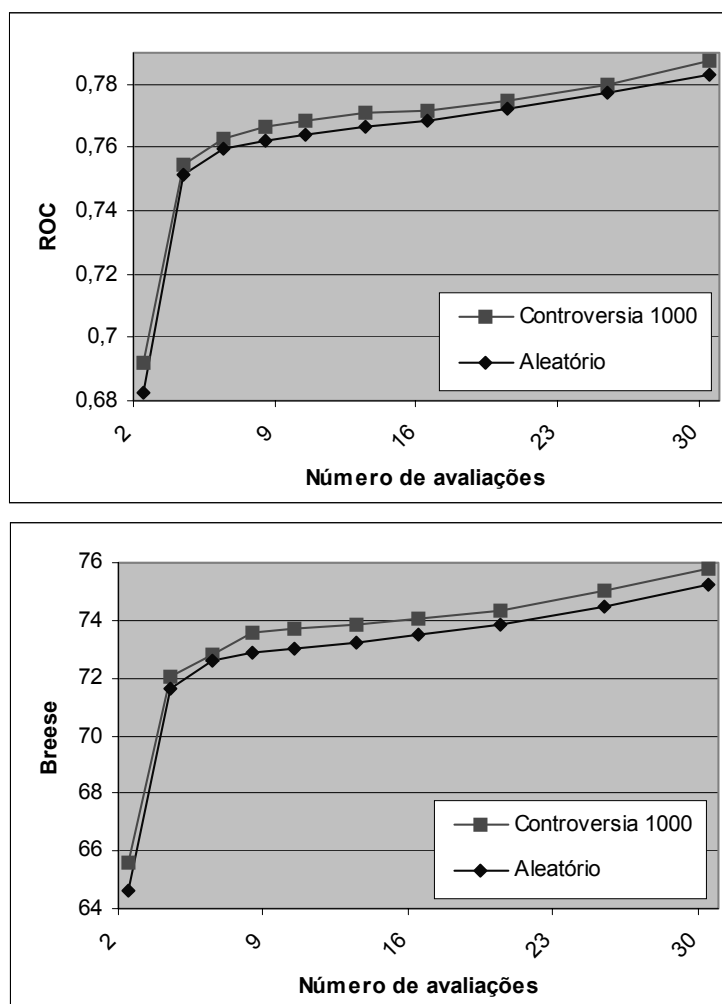


Figura 6.9 – Gráficos da performance das predições usando as avaliações dos itens selecionados com a controvérsia medida em uma amostra de 1000 usuário e aleatoriamente.

ROC	Controvérsia 1000		Aleatório		<i>p</i>	Significativo ($\alpha = 0,1$)
Número de avaliações	X₁	S₁	X₂	S₂		
2	0,69195	0,10970	0,68244	0,07723	0,01	Sim
4	0,75440	0,10072	0,75126	0,09358	0,23	Não
6	0,76250	0,09897	0,75981	0,09692	0,27	Não
8	0,76685	0,10108	0,76193	0,09806	0,13	Não
10	0,76855	0,10063	0,76380	0,09800	0,14	Não
13	0,77082	0,10077	0,76636	0,09771	0,16	Não
16	0,77156	0,10245	0,76824	0,09757	0,23	Não
20	0,77484	0,10250	0,77206	0,09925	0,27	Não
25	0,78002	0,10199	0,77763	0,09820	0,30	Não
30	0,78766	0,09848	0,78301	0,09727	0,14	Não

Tabela 6.2 – Tabela da performance da predições segunda a métrica ROC utilizando as avaliações dos itens selecionados com a controvérsia e aleatoriamente.

Breese	Controvérsia 1000		Aleatório		<i>p</i>	Significativo ($\alpha = 0,1$)
Número de avaliações	X₁	S₁	X₂	S₂		
2	65,583	13,179	64,594	10,403	0,03	Sim
4	72,049	11,398	71,598	10,808	0,18	Não
6	72,828	11,144	72,608	10,810	0,33	Não
8	73,539	10,930	72,886	10,708	0,09	Sim
10	73,739	10,851	72,987	10,626	0,06	Sim
13	73,863	10,503	73,250	10,385	0,09	Sim
16	74,049	10,490	73,493	10,398	0,12	Não
20	74,331	10,439	73,877	10,473	0,17	Não
25	74,995	10,381	74,478	10,322	0,13	Não
30	75,768	10,157	75,211	10,101	0,11	Não

Tabela 6.3 – Tabela da performance das predições segunda a métrica Breese utilizando as avaliações dos itens selecionados com a controvérsia e aleatoriamente.

A comparação entre as metodologias mostra que apesar da seleção com controvérsia ser em média melhor do que a seleção aleatória, ainda não é possível provar a superioridade dessa técnica de seleção utilizando um teste de hipótese a um nível de 10%. Na tabela 6.2, utilizando a métrica ROC, apenas no caso onde são selecionadas 2 avaliações pode-se concluir que a seleção com controvérsia é melhor do que uma seleção aleatória. Na tabela 6.3, utilizando a métrica Breese, além do caso de 2 avaliações selecionadas, também nos casos entre 8 e 13 itens selecionados pode-se concluir que a controvérsia foi melhor.

Seleção baseada na popularidade

A utilização da popularidade como critério de seleção das avaliações também foi testada. A popularidade testada aqui é global (popularidade total), ou seja, a popularidade de um item será o número de avaliações que ele recebeu no conjunto de usuários do sistema. A figura 6.10 mostra o gráfico do desempenho dessa metodologia na seleção de avaliações comparada com o método aleatório e o da controvérsia com amostra de 1000.

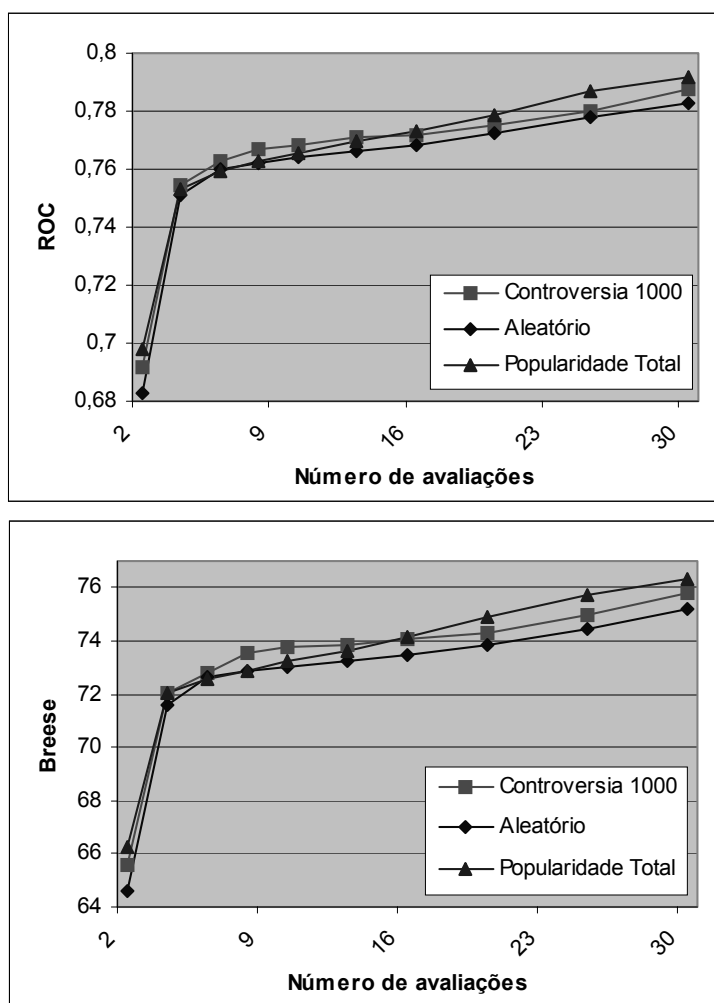


Figura 6.10 – Gráficos da performance das predições usando as avaliações dos itens selecionados com a controvérsia medida em uma amostra de 1000 usuários, com a popularidade total e aleatoriamente.

	ROC			Breese		
Número de avaliações	Controvérsia 1000	Popularidade Total	Aleatória	Controvérsia 1000	Popularidade Total	Aleatória
2	0,69195	0,69824	0,68244	65,583	66,271	64,594
4	0,75440	0,75307	0,75126	72,049	72,032	71,598
6	0,76250	0,75930	0,75981	72,828	72,547	72,608
8	0,76685	0,76247	0,76193	73,539	72,832	72,886
10	0,76855	0,76585	0,76380	73,739	73,232	72,987
13	0,77081	0,76999	0,76636	73,863	73,592	73,250
16	0,77156	0,77324	0,76824	74,049	74,110	73,493
20	0,77484	0,77864	0,77206	74,331	74,877	73,877
25	0,78002	0,78669	0,77763	74,995	75,729	74,478
30	0,78766	0,79164	0,78301	75,768	76,287	75,211

Tabela 6.4 – Tabela que mostra os valores médios de ROC e Breese para as metodologia de seleção com controvérsia em uma amostra de 1000 usuários, com a popularidade total e aleatória.

O gráfico da figura 6.10 mostra que a performance das predições utilizando as avaliações selecionadas, segundo o critério de popularidade, tem uma performance inferior ao método de seleção de avaliações utilizando a controvérsia quando são selecionados um número inferior a 16 itens avaliados. Entretanto, essa metodologia mostrou-se melhor quando selecionado um número maior de itens. A comparação da performance média de cada metodologia comparada no gráfico da figura 6.10 é mostrada na tabela 6.4.

Quando feita a comparação entre os métodos aleatório e de seleção com popularidade, observa-se que a popularidade como critério de seleção é significativamente superior à seleção aleatória quando o número de itens selecionadas é grande. As tabelas 6.5 e 6.6 mostram a comparação entre as duas metodologias segundo as métricas de performance da predição ROC e Breese.

ROC	Popularidade Total		Aleatório		<i>p</i>	Significativo ($\alpha = 0,1$)
Número de avaliações	X₁	S₁	X₂	S₂		
2	0,69824	0,11533	0,68244	0,07723	0,00	Sim
4	0,75307	0,10459	0,75126	0,09358	0,34	Não
6	0,75930	0,10413	0,75981	0,09692	0,55	Não
8	0,76247	0,10368	0,76193	0,09806	0,45	Não
10	0,76585	0,10279	0,76380	0,09800	0,32	Não
13	0,76999	0,10000	0,76636	0,09771	0,21	Não
16	0,77324	0,09933	0,76824	0,09757	0,13	Não
20	0,77864	0,09741	0,77206	0,09925	0,07	Sim
25	0,78669	0,09712	0,77763	0,09820	0,02	Sim
30	0,79164	0,09502	0,78301	0,09727	0,02	Sim

Tabela 6.5 – Tabela da performance das predições segunda a métrica ROC utilizando as avaliações dos itens selecionados com a popularidade total e aleatoriamente.

Breese	Popularidade Total		Aleatório		<i>p</i>	Significativo ($\alpha = 0,1$)
Número de avaliações	X₁	S₁	X₂	S₂		
2	66,271	13,074	64,594	10,403	0,00	Sim
4	72,032	11,293	71,598	10,807	0,19	Não
6	72,547	11,400	72,608	10,810	0,55	Não
8	72,832	11,333	72,886	10,708	0,54	Não
10	73,232	11,065	72,987	10,626	0,31	Não
13	73,592	10,843	73,250	10,385	0,24	Não
16	74,110	10,852	73,493	10,398	0,10	Sim
20	74,877	10,491	73,877	10,473	0,02	Sim
25	75,729	10,218	74,478	10,322	0,00	Sim
30	76,287	9,913	75,211	10,101	0,01	Sim

Tabela 6.6 – Tabela da performance das predições segunda a métrica Breese utilizando as avaliações dos itens selecionados com a popularidade total e aleatoriamente.

A performance da metodologia de seleção baseada na popularidade dos itens mostrou-se superior apenas quando uma quantidade de avaliações maior do que 20 é selecionada. No entanto, foi possível observar que a popularidade de um item é relevante no momento de selecioná-lo.

6.4.6 ActiveCP

Apesar de existirem evidências de que a controvérsia e a popularidade de um item são critérios relevantes no momento de selecionar os itens avaliados por um usuário, as metodologias propostas em cada um desses critérios foram capazes de selecionar os itens que resultaram em melhores previsões apenas quando determinadas quantidades de itens são selecionadas.

A seleção dos itens baseado na sua controvérsia não teve uma performance significativamente melhor do que a seleção aleatória. Isto aconteceu na maioria dos casos, quando as previsões são avaliadas pela métrica de seleção ROC. Quando as previsões são avaliadas pela métrica Breese, essa metodologia de seleção foi significativamente melhor do que a aleatória apenas em alguns pontos.

A seleção dos itens baseado na sua popularidade mostrou performance significativamente melhor do que a seleção aleatória apenas quando uma quantidade superior a aproximadamente 20 itens foi selecionada.

A controvérsia e a popularidade parecem ser características ortogonais de um item dentro do sistema. Um item pode ser considerado controvertido sendo ou não popular. Ao mesmo tempo um item pode ser considerado popular sendo ou não controvertido. A tabela 6.7 mostra exemplos extraídos da base de avaliações dos usuários para filmes:

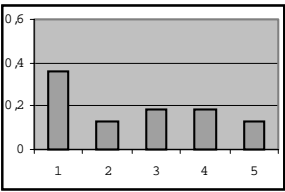
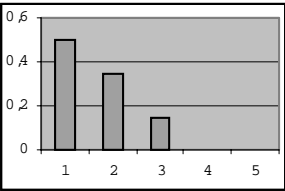
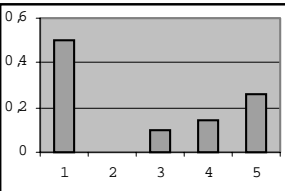
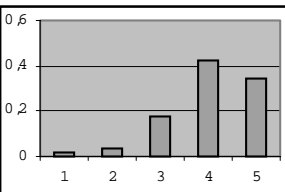
Filme	Distribuição das avaliações	Variância das avaliações	Número de avaliações
Assassinos por natureza		2,12	14.759
Um favor inocente		0,52	20
Bye-Bye		3,06	25
O Fugitivo		0,78	24.278

Tabela 6.7 – Exemplos de filmes da base de dados e a distribuição, variância das avaliações e número de avaliações para cada um

A tabela 6.7 mostra exemplos de filmes extraídos da base de dados completa. A distribuição e variância das avaliações e o número de avaliações dos usuários do sistema são mostrados. O filme “Assassinos por natureza” mostrou-se bastante popular e controvertido. O filme “Um favor inocente”, por sua vez, não teve muitas avaliações e não é muito controvertido. O filme “Bye-Bye” é pouco popular mas controvertido, e o filme “O Fugitivo” é bastante popular, ao mesmo tempo que a maioria dos usuários concorda quanto à sua avaliação. Observamos, portanto, que existem na base de dados filmes com características de popularidade e controvérsia bastante independentes.

Sugerimos o método de seleção *ActiveCP* capaz de combinar esses dois critérios no momento de selecionar os itens. No nosso método são produzidas duas ordenações dos itens do sistema, uma ordenação dos itens pela controvérsia, do mais ao menos controvertido, e outra ordenação crescente pela popularidade, ou seja, do mais ao menos popular.

A cada item são associados dois valores que indicam sua ordem de preferência dentre os itens do sistema segundo cada um dos critérios. Para um item i ordenado na posição p_i em uma lista com n itens, o valor v_i associado a esse item é calculado segundo a equação 6.3:

$$v_i = \frac{p_i}{1-n} - \frac{n}{1-n} \quad (\text{Eq. 6.3})$$

A equação 6.3 mapeia a posição p_i de um item em uma determinada ordem para um valor $v_i \in [0,1]$, de forma que o primeiro item da ordem ($p_i = 1$) é mapeado para um valor $v_i = 1$, e o último item da ordem ($p_i = n$) é mapeado para um valor $v_i = 0$. Os dois valores associados a um item serão os valores mapeados das ordens obtidas segundo os critérios de popularidade e controvérsia. Para um item i , os valores vc_i e vp_i são obtidos pelo mapeamento das posições desse item nas ordenações feitas respectivamente pela controvérsia e popularidade, utilizando a equação 6.3. Os valores vc_i e vp_i obtidos são combinados para obter um único valor V_i , como segue na equação 6.4:

$$V_i = w_c vc_i + w_p vp_i \quad (\text{Eq. 6.4})$$

Na equação 6.4, w_c é o peso dado à controvérsia de um item e w_p o peso dado à popularidade de um item. Os valores de w_c e w_p vão determinar o quanto cada um dos critérios irá influenciar no valor V_i calculado para cada item.

O critério V_i é chamado de CP (Controvérsia - Popularidade) por ser um critério que combina a controvérsia e a popularidade dos itens em um único valor. O nosso método *ActiveCP* consiste em selecionar os itens que tiverem o maior valor do critério CP utilizado.

Para testar o método de seleção ActiveCP, os pesos w_c e w_p são variados de forma que a controvérsia ou a popularidade tivessem maior ou menor influência na seleção das avaliações. As combinações de pesos testados são indicadas na tabela 6.8.

	CP 0	CP 1	CP 2	CP 3	CP 4	CP 5	CP 6	CP 7	CP 8	CP 9	CP 10
w_c	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0
w_p	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0

Tabela 6.8 – Tabela dos pesos utilizados no critério CP de seleção das avaliações.

A variação feita com os pesos tem o propósito de verificar a performance das predições do sistema usando as avaliações dos itens selecionados segundo um critério baseado apenas na controvérsia (CP 0), diminuindo gradualmente a influência da controvérsia e, ao mesmo tempo, aumentando a influência da popularidade, até chegar a um critério baseado apenas na popularidade (CP 10).

O método ActiveCP foi testado utilizando uma amostra de 1000 usuários para calcular a controvérsia e a popularidade total. As tabelas 6.9 e 6.10 mostram os resultados da seleção das predições com as métricas ROC e Breese respectivamente, usando as avaliações dos itens selecionados pelo critério CP.

	CP 0	CP 1	CP 2	CP 3	CP 4	CP 5	CP 6	CP 7	CP 8	CP 9	CP 10
2	0,6920	0,6924	0,6940	0,6970	0,7012	0,7033	0,7007	0,6990	0,6992	0,6969	0,6982
4	0,7544	0,7559	0,7564	0,7571	0,7561	0,7582	0,7560	0,7537	0,7514	0,7522	0,7531
6	0,7625	0,7643	0,7637	0,7653	0,7642	0,7664	0,7626	0,7613	0,7611	0,7589	0,7593
8	0,7669	0,7670	0,7672	0,7673	0,7665	0,7677	0,7642	0,7639	0,7642	0,7634	0,7625
10	0,7686	0,7696	0,7691	0,7680	0,7680	0,7702	0,7675	0,7661	0,7662	0,7651	0,7659
13	0,7708	0,7705	0,7700	0,7705	0,7701	0,7721	0,7695	0,7698	0,7713	0,7698	0,7700
16	0,7716	0,7724	0,7727	0,7721	0,7731	0,7744	0,7737	0,7740	0,7752	0,7749	0,7732
20	0,7748	0,7759	0,7758	0,7779	0,7779	0,7816	0,7812	0,7803	0,7802	0,7794	0,7786
25	0,7800	0,7810	0,7820	0,7838	0,7848	0,7878	0,7882	0,7873	0,7862	0,7862	0,7867
30	0,7877	0,7870	0,7901	0,7906	0,7930	0,7927	0,7938	0,7938	0,7926	0,7914	0,7916
Área	21,550	21,570	21,582	21,605	21,618	21,678	21,643	21,623	21,619	21,570	21,595

Tabela 6.9 – Essa tabela mostra a performance das predições com a métrica ROC quando os itens são selecionados segundo o critério CP com diferentes pesos. Para cada número de avaliações selecionada está marcado de negrito o critério que teve melhor performance média. Na última linha da tabela se encontra a área sob a curva formada pelos pontos (performance para número de avaliações selecionadas) conhecidos.

	CP 0	CP 1	CP 2	CP 3	CP 4	CP 5	CP 6	CP 7	CP 8	CP 9	CP 10
2	65,584	65,867	65,901	66,367	66,573	67,200	66,614	66,344	66,632	66,179	66,271
4	72,050	72,330	72,356	72,345	72,246	72,788	72,415	71,970	71,837	71,952	72,033
6	72,828	73,255	73,169	73,399	73,406	73,623	73,070	72,809	72,772	72,525	72,547
8	73,539	73,537	73,498	73,517	73,522	73,619	73,121	73,071	73,079	72,907	72,833
10	73,739	73,695	73,597	73,500	73,561	73,930	73,515	73,332	73,292	73,199	73,233
13	73,864	73,812	73,784	73,868	73,808	74,138	73,771	73,668	73,791	73,619	73,593
16	74,049	74,072	74,087	74,089	74,230	74,335	74,236	74,256	74,279	74,380	74,110
20	74,332	74,552	74,370	74,694	74,809	75,046	74,974	75,040	75,002	74,955	74,878
25	74,996	75,077	75,027	75,366	75,459	75,839	75,807	75,766	75,725	75,708	75,730
30	75,769	75,746	75,956	76,029	76,376	76,445	76,425	76,418	76,416	76,222	76,288
Área	2066,4	2069,3	2068,3	2072,6	2074,9	2082,69	2076,2	2073,8	2073,8	2071,5	2070,7

Tabela 6.10 – Essa tabela mostra a performance das predições com a métrica Breese quando os itens são selecionadas segundo o critério CP com diferentes pesos. Para cada número de avaliações selecionada está marcado de negrito o critério que teve melhor performance média. Na última linha da tabela se encontra a área sob a curva formada pelos pontos (performance para número de avaliações selecionadas) conhecidos.

Os resultados mostrados nas tabelas 6.9 e 6.10 são avaliados para cada número de itens selecionados. O critério que obteve maior performance média para cada ponto se encontra em negrito. Os valores pontuais talvez não sejam conclusivos quanto a qual critério é o melhor. A área sob a curva do desempenho por número de itens avaliados selecionados é mostrada na última linha como forma de resumir a performance de cada critério em um único valor. Tanto para a métrica ROC quanto Breese, quando levada em consideração a área da curva de desempenho, o critério CP 5 obteve o maior valor.

Observa-se que os critérios da controvérsia e popularidade, quando usados isoladamente, mostrados em CP 0 onde usa-se apenas a controvérsia, e em CP 10 que utiliza somente a popularidade, são quase sempre em média pior do que quando combinamos os dois critérios. O método CP 5, que obteve maior área da curva de desempenho, é o que atribui pesos iguais tanto para a influencia da controvérsia quanto para a da popularidade.

A figura 6.11 mostra o gráfico do desempenho da metodologia de seleção com o critério CP 5, comparando com a seleção aleatória de avaliações.

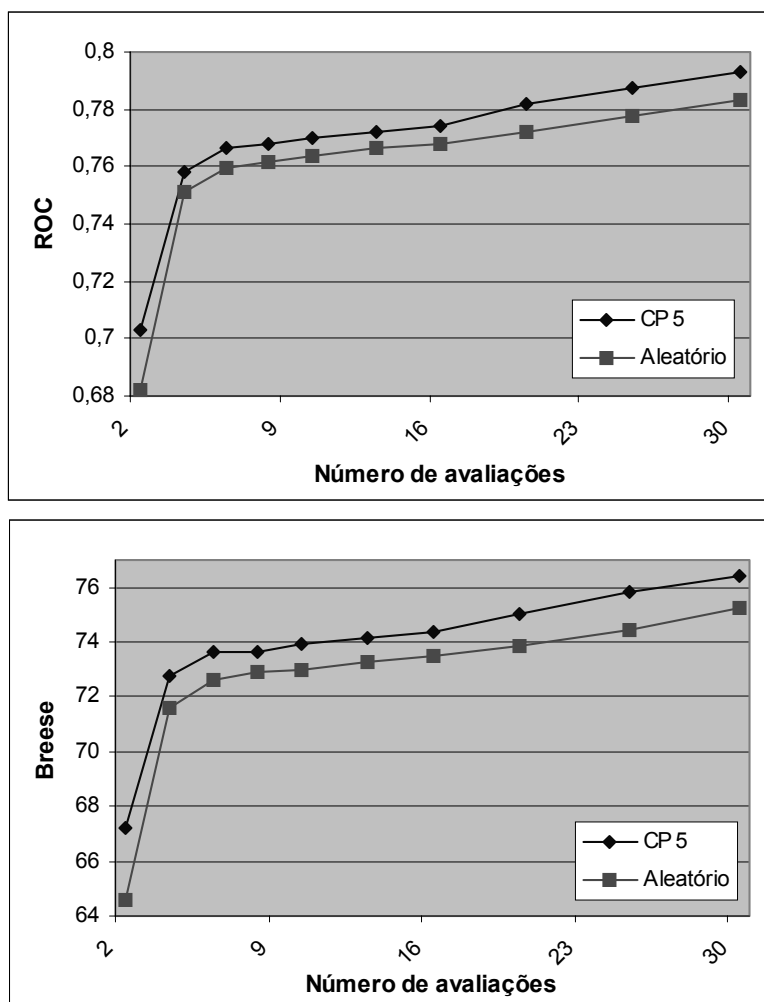


Figura 6.11 – Gráficos comparando as metodologia de seleção com o critério CP e aleatória.

ROC	CP 5		Aleatório		p	Significativo ($\alpha = 0,1$)
Número de avaliações	x_1	s_1	x_2	s_2		
2	0,70328235	0,112196	0,68244	0,077232	0,000	Sim
4	0,758189407	0,101586	0,751268	0,093589	0,057	Sim
6	0,766431332	0,099787	0,759812	0,096926	0,066	Sim
8	0,767666859	0,099094	0,761933	0,098061	0,097	Sim
10	0,770222509	0,100638	0,7638	0,098008	0,074	Sim
13	0,772134089	0,100282	0,766361	0,097719	0,096	Sim
16	0,774372875	0,099758	0,768243	0,097572	0,082	Sim
20	0,78162038	0,09962	0,772069	0,099256	0,016	Sim
25	0,787758579	0,096624	0,77763	0,0982	0,010	Sim
30	0,792748385	0,096323	0,783016	0,097271	0,012	Sim

Tabela 6.11 – Tabela da performance das predições segunda a métrica ROC utilizando as avaliações dos itens selecionados com o CP 5 e aleatoriamente.

Breese	CP 5		Aleatório		p	Significativo ($\alpha = 0,1$)
Número de avaliações	x_1	s_1	x_2	s_2		
2	67,20039799	12,57011	64,594	10,40312	0,000	Sim
4	72,78837383	10,62759	71,59879	10,8078	0,007	Sim
6	73,62264997	10,35276	72,60887	10,81094	0,016	Sim
8	73,61861591	10,3018	72,88648	10,70897	0,060	Sim
10	73,92990867	10,40611	72,98726	10,62624	0,023	Sim
13	74,13782229	10,40674	73,25022	10,38577	0,028	Sim
16	74,33502013	10,40232	73,4937	10,39826	0,035	Sim
20	75,04619619	10,3166	73,87777	10,47393	0,006	Sim
25	75,8387234	10,01953	74,47875	10,32246	0,001	Sim
30	76,44541011	9,835994	75,21148	10,10114	0,003	Sim

Tabela 6.12 -Tabela da performance das predições segunda a métrica ROC utilizando as avaliações dos itens selecionados com o CP 5 e aleatoriamente.

O gráfico da figura 6.11 mostra que o desempenho médio da metodologia de seleção baseada no critério CP 5 é sempre superior ao desempenho da seleção aleatória. As tabelas 6.11 e 6.12 mostram os valores dos gráficos da figura 6.11 e o teste de significância da diferença do desempenho médio das duas métricas. O desempenho médio da seleção com o critério CP 5 mostrou-se superior ao desempenho da seleção aleatória ao nível de 10 %.

6.4.7 Análise do método ActiveCP

O estudo das avaliações dos usuários para os itens e de sua distribuição através de critérios como a controvérsia e a popularidade se mostrou adequado para o problema de seleção dos itens que o usuário avalia no desenvolvimento de um método de *selective sampling* para FI com vizinhos mais próximos. Cada um desses critérios, quando utilizado de forma independente, mostrou um ganho no desempenho médio das predições feitas utilizando as avaliações selecionadas quando comparada à seleção aleatória. Entretanto, essa diferença não foi estatisticamente significativa na maioria dos casos, o que dificulta justificar a sua utilização. Somente quando foram combinados no método ActiveCP de seleção, integrando popularidade e controvérsia, foi possível alcançar um desempenho médio

significativamente superior à seleção aleatória. A figura 6.12 mostra o gráfico que permite comparar as metodologias de seleção com a controvérsia, popularidade e a combinação das duas através do critério CP 5.

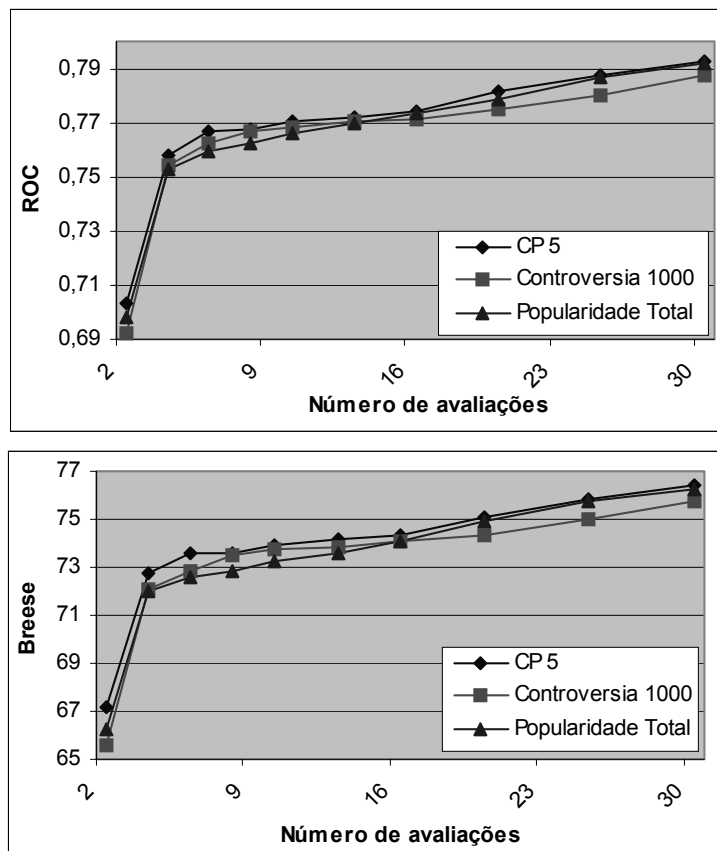


Figura 6.12 – Gráfico comparando as metodologias de seleção baseadas na controvérsia, popularidade e na combinação das duas através de CP 5.

A combinação da controvérsia e popularidade através do método ActiveCP com o critério CP 5 teve desempenho médio igual ou superior a quando apenas a controvérsia ou a popularidade foi utilizada para qualquer número de itens selecionados. Essa observação leva à conclusão que um item é mais relevante para o perfil de um usuário quando ele é ao mesmo tempo popular e controvertido dentro da comunidade.

É importante também observar melhor o ganho obtido utilizando uma aprendizagem ativa do perfil do usuário. Nota-se que o ganho de performance nas predições aumenta gradualmente com o acréscimo no número de avaliações feitas pelo usuário. O aumento é grande apenas quando efetuadas as 5 primeiras

avaliações. A partir deste ponto, esse aumento é bem menor quando mais itens são avaliados. Quando as avaliações são selecionadas através da metodologia CP 5, há um sensível ganho de performance que pode ser melhor apreciado quando observado o número de avaliações necessárias para atingir uma performance desejada na qualidade das predições. A figura 6.13 mostra o gráfico comparando quantas avaliações seriam necessárias para atingir um desempenho desejado com as métricas ROC e Breese, quando essas avaliações são selecionadas aleatoriamente ou utilizando o critério CP 5.

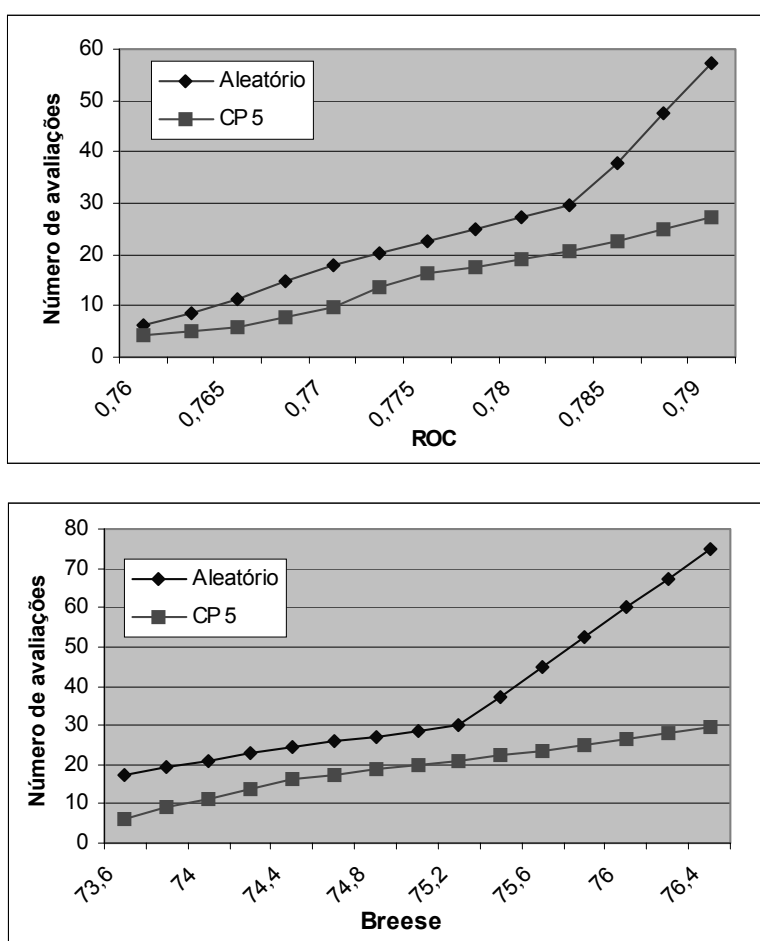


Figura 6.13 – Gráficos com o número de avaliações necessárias para obter a performance desejada utilizando a seleção aleatória das avaliações e a seleção pelo critério CP 5.

Os gráficos da figura 6.13 mostram que a seleção das avaliações pelo critério CP 5 permite alcançar uma qualidade de predições igual àquela quando as avaliações são feitas aleatoriamente, mas com um número bem menor de avaliações.

Para desenhar os gráficos da figura 6.13 a seleção aleatória de avaliações foi estendida além das 30 avaliações selecionadas para o experimento chegando ao máximo de 80 avaliações, quando todas as avaliações do usuário foram selecionadas. A performance do caso onde todas as 80 avaliações dos usuários foram selecionadas e utilizadas pelo algoritmo de predição foi similar àquela do caso onde foram selecionadas 30 avaliações pelo critério CP 5. A tabela 6.13 mostra esse resultado:

	CP 5		Perfil Completo		<i>p</i>	Significativo ($\alpha = 0,1$)
	30 avaliações		80 avaliações			
	x ₁	s ₁	x ₂	s ₂		
ROC	0,7927	0,0963	0,7957	0,0878	0,769	Não
Breese	76,445	9,836	76,530	9,665	0,577	Não

Tabela 6.13 – Comparação da performance das predições quando são usadas as avaliações de 30 itens selecionados pelo critério CP 5 e o perfil completo do usuário contendo todas as suas 80 avaliações.

Nas métricas Breese e ROC, a performance das predições utilizando 30 avaliações selecionadas pelo critério CP 5 tem performance semelhante a quando é utilizado todo o conjunto de avaliações de cada usuário, num total de 80 avaliações. A tabela mostra também que a diferença da performance média dos dois não é estatisticamente significativa ao nível de 10%.

A figura 6.14 mostra os gráficos com a razão N_a/N_{CP5} para diferentes valores das métricas ROC e Breese, onde N_a é o número de itens selecionados aleatoriamente necessário para obter a performance indicada e N_{CP5} o número de itens selecionados usando o critério CP 5 para obter a mesma performance.

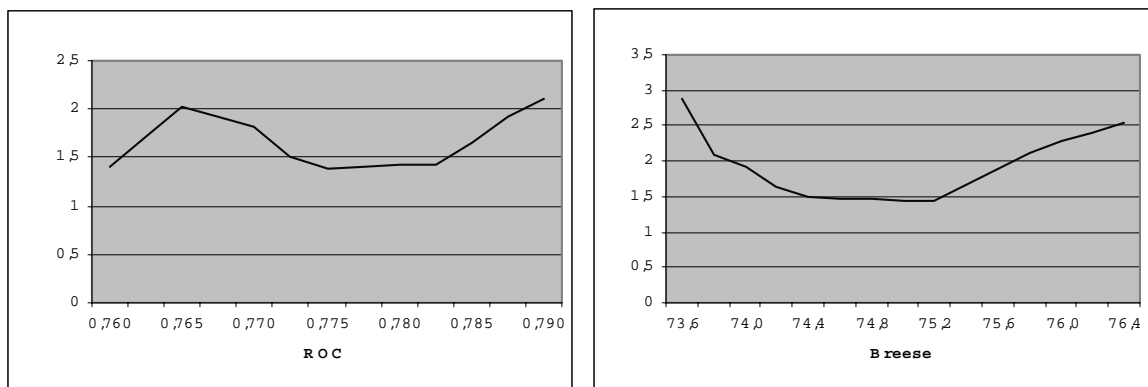


Figura 6.14 – Razão N_a/N_{CP5} entre o número de itens selecionados aleatoriamente N_a e selecionados pelo critério CP 5 para obter uma performance desejada nas métricas ROC e Breese

Esse gráfico mostra a proporção de itens selecionados aleatoriamente que equivalem aos itens selecionados pelo critério CP 5 para alcançar a mesma performance. Na métrica ROC essa proporção varia entre 1,5 e 2 indicando que são necessárias de 50% até o dobro de avaliações aos itens selecionados aleatoriamente para obter a mesma performance de quando são avaliados itens selecionados pelo critério CP 5. Na métrica Breese essa proporção é ainda maior, chegando a 2,5 vezes ou 3 vezes em alguns casos.

A figura 6.15 mostra os gráficos que comparam quantas avaliações devem ser selecionadas aleatoriamente para obter a mesma performance de predição, quando as avaliações são selecionadas segundo o critério CP 5.

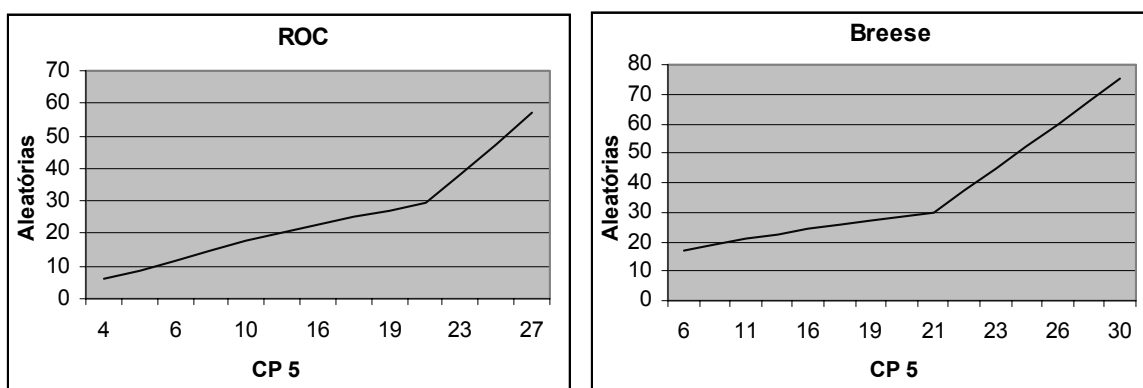


Figura 6.15 – Quantidade de avaliações que devem ser selecionadas aleatoriamente para obter performance média de predições igual à obtida pela quantidade de avaliações selecionadas pelo critério CP 5.

Pode-se concluir através dos gráficos da figura 6.15 que o número de avaliações selecionadas pelo critério CP 5 sempre corresponde a uma quantidade maior de avaliações selecionadas aleatoriamente. Além do mais, essa diferença tende a aumentar quando o número de avaliações é maior, conforme mostram os gráficos expostos acima.

6.5 Conclusão

Nesse capítulo foi introduzido o problema da aquisição do perfil do usuário em um sistema de FI. A aprendizagem ativa foi apresentada como solução para problemas de aprendizagem de máquina nos quais a aquisição de exemplos para o aprendizado é dispendiosa, possivelmente por exigir esforço humano, como é o caso da aprendizagem do perfil em sistemas de FI através de avaliações explícitas.

Nesse capítulo foi desenvolvida uma metodologia de seleção das avaliações capaz de tornar mais eficiente a aquisição do perfil do usuário. A controvérsia e a popularidade de um item foram identificados como atributos relevantes na seleção das avaliações dos usuários. Para calcular a controvérsia de um item foi utilizada a variância das avaliações feitas para esse item. A controvérsia calculada em uma amostra de até 1000 usuários que avaliaram o item, apresentou melhor performance média das predições feitas com as avaliações dos itens selecionados do que a seleção baseada na controvérsia calculada em toda a comunidade ou em tamanhos diferentes de amostras.

Tanto a seleção de avaliações baseada na controvérsia quanto na popularidade apresentaram desempenho médio das predições superiores à seleção aleatória das avaliações. Entretanto, o teste de hipótese mostrou que a diferença entre as médias não era significativa na maior parte dos casos de seleção de números diferentes de avaliações.

A solução encontrada foi utilizar o método ActiveCP que combina os critérios da controvérsia e popularidade em um único critério capaz de levar em consideração os dois fatores no momento da seleção. O critério CP (Controvérsia-Popularidade)

desenvolvido combina os dois critérios através da ordenação dos itens segundo cada um dos critérios, favorecendo aqueles itens que, ao mesmo tempo, estão entre os mais controvertidos e mais populares. Foi variada a influência da popularidade e da controvérsia no critério CP para verificar que o critério CP 5, onde a influência dos dois é a mesma, apresenta os melhores resultados. A performance das predições com as avaliações selecionadas segundo o critério CP 5 foi comparada com as avaliações selecionadas aleatoriamente para verificar que o método ActiveCP com o critério CP 5 é capaz de selecionar melhor as avaliações. A diferença da performance média foi verificada significativa ao nível de 10%.

Finalmente foi mostrado que a seleção das avaliações utilizando o critério CP 5 é capaz de diminuir bastante o número de avaliações necessárias para atingir um nível de performance desejado. Também foi mostrado que a performance do algoritmo de predição usando as avaliações de determinado número de itens selecionadas pelo critério CP 5 equivale à performance do algoritmo com um número bastante superior de avaliações de itens selecionados aleatoriamente.

Capítulo 7

*GEP: Um sistema de recomendação para
entretenimento*

Nesse capítulo é descrito o sistema de recomendação *GEP* (Guia de Entretenimento Personalizado), que foi desenvolvido com a finalidade de mostrar a aplicação da metodologia de filtragem de informação e de aprendizagem ativa do perfil do usuário.

7.1 Visão geral do sistema

O sistema *GEP* foi desenvolvido numa parceria entre o Centro de Informática da UFPE e o CESAR (Centro de Estudos e Sistemas Avançados do Recife). O objetivo do sistema é mostrar a aplicação da tecnologia desenvolvida nesse trabalho em um sistema de recomendação.

O sistema *GEP* foi idealizado como um sistema capaz de ajudar um usuário a escolher uma opção de entretenimento dentre uma quantidade grande de opções disponíveis. Para isso o *GEP* é capaz de aprender sobre o perfil do usuário através de avaliações explícitas dos itens, e utilizando o conhecimento que possui sobre ele, gerar um guia com as opções de entretenimento de seu maior interesse. O sistema é portanto um sistema de informação sobre eventos de entretenimento, com um cadastro dos eventos disponíveis e a data e hora de sua realização, onde o usuário pode tanto recuperar e consultar os eventos pelo local, data e categoria, como também terá a opção de avaliar eventos e receber sugestões de eventos segundo o seu perfil.

A tecnologia usada no sistema para aquisição do perfil do usuário e geração de recomendações é a mesma descrita nesta dissertação. A mesma tecnologia pode ser utilizada também para personalizar o acesso a informações sobre livros, CDs, notícias, artigos, etc. A escolha da sua aplicação ao domínio de eventos de entretenimento deve-se ao fato desse domínio apresentar a característica de possuir grande quantidade de opções e usuários com gostos bastante variados. Verifica-se portanto uma necessidade de personalização dos guias de eventos.

O tipo de evento de entretenimento que pode ser recomendado pode variar bastante. Foi desenvolvida uma ontologia de eventos que podem ser recomendados pelo GEP. A ontologia é mostrada na figura 7.1

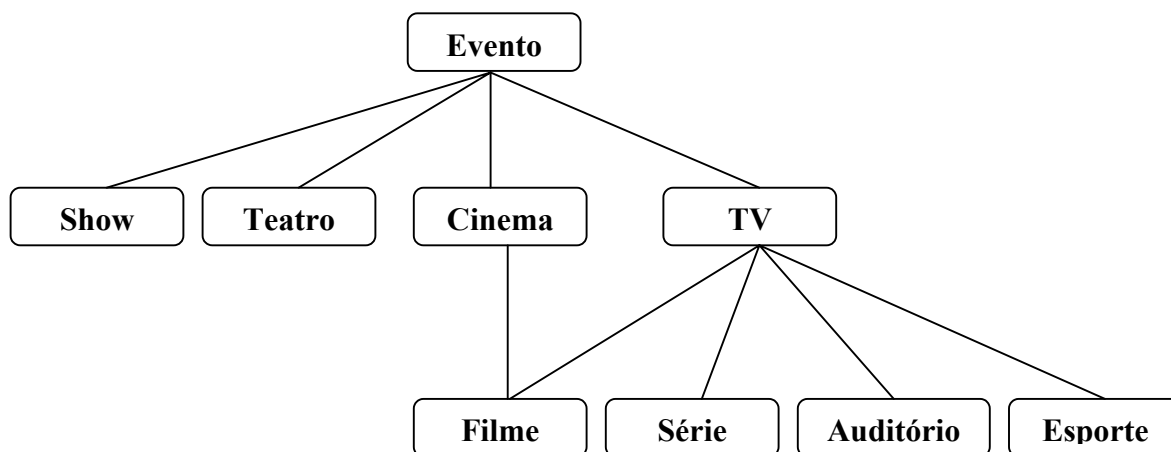


Figura 7.1- Ontologia de eventos de entretenimento

Para o protótipo descrito nessa dissertação foi considerada apenas a parte da ontologia que trata de eventos de TV. Entretanto, a extensão do sistema para abranger toda a ontologia da figura 7.1 não apresenta dificuldades maiores. A decisão de restringir o sistema se deve principalmente à dificuldade de obter dados sobre tais eventos.

7.2 Ambiente de desenvolvimento

O suporte necessário para o desenvolvimento do GEP foi fornecido pelo CESAR e pelo CIn-UFPE. O desenvolvimento do sistema teve a participação de 4 pessoas além do autor deste trabalho.

A linguagem de programação utilizada foi Java (com o compilador *JDK 1.2.2*). Escolheu-se a linguagem Java devido à sua portabilidade uma vez que existem implementações do seu interpretador para diferentes plataformas, tornando o próprio sistema portátil. Também o fato de Java ser uma linguagem orientada a objeto com uma API (*Application Program Interface*) bastante completa facilitou bastante o desenvolvimento.

Como sistema gerenciador de banco de dados foi utilizada a versão 7.2 do *DB2 Universal Edition*. A comunicação da aplicação Java com o servidor de dados é realizada através da API Java JDBC (*Java Database Connectivity*), que trata de todas as questões de troca de dados entre aplicação e SGBD.

A comunicação entre o usuário e o sistema é feita através de um *browner* que se comunica com o servidor http e de aplicação Jakarta Tomcat versão 3.3. No servidor de aplicação utilizado, as requisições do usuário são tratadas pelo sistema utilizando Java Servlet [Servlet 02] e o resultado é retornado ao usuário através de páginas geradas dinamicamente via JSP (*Java Server Pages*) [JSP 02].

Essas e outras ferramentas utilizadas no desenvolvimento do protótipo são resumidos na tabela 7.1.

Ferramenta	Descrição
DB2 Universal Edition 7.2	SGBD utilizado para armazenar os dados persistentes.
Visual Age for Java 4.0	Ambiente de desenvolvimento Java com construção para o JDK 1.2.2.
Rational Rose 2000	Para a modelagem do sistema baseado no padrão UML (<i>Unified Modelling Language</i>).
Jakarta Tomcat 3.3	Servidor Web e de aplicação responsável pela comunicação entre usuário e sistema.
Adobe Photoshop 6.0	Ambiente de criação e manipulação de imagens.
Macromedia Dreamweaver 4	Ambiente de desenvolvimento de interface <i>Web</i> .

Tabela 7.1 – Ferramentas utilizadas no desenvolvimento do sistema GEP

7.3 Projeto e implementação

Nesta seção é descrito o projeto e a implementação do sistema GEP. A figura 7.2 mostra um modelo geral do sistema.

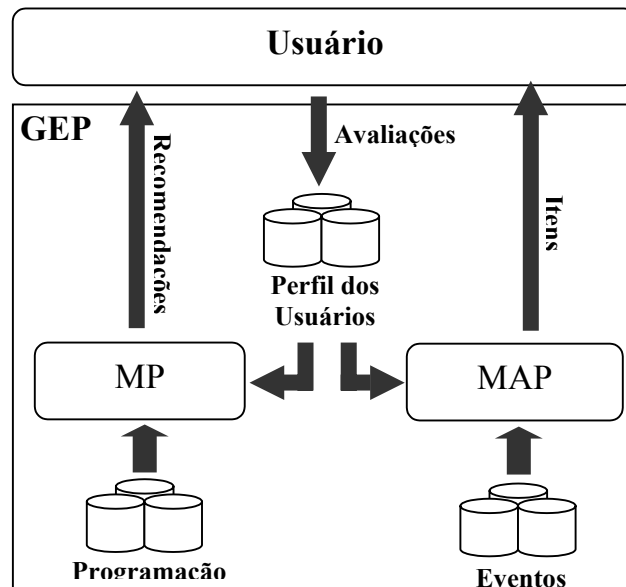


Figura 7.2 – Modelo geral do sistema GEP

O cliente do GEP acessa o sistema para obter informações sobre os eventos cadastrados no sistema. A qualquer momento o usuário pode consultar a programação de eventos como em um sistema de informação comum. O usuário poderá também fornecer avaliações sobre os eventos que serão usados para formar o seu perfil. Além disso o sistema interage com o usuário através dos seguintes módulos:

Módulo de Personalização (MP)

É a parte do sistema responsável pela geração de recomendações para o usuário. O MP recebe a programação dos eventos e o perfil do usuário e retorna um guia de programação com os eventos de maior interesse do usuário segundo o seu perfil armazenado.

Módulo de Aquisição de Perfil (MAP)

É a parte do sistema responsável pela aquisição do perfil do usuário. O MAP recebe uma coleção de eventos e o perfil do usuário (se já disponível), e retorna um conjunto de eventos que, uma vez avaliados pelo usuário, ajudam na construção do seu perfil de forma eficiente.

7.4 Arquitetura do sistema

No projeto de implementação do modelo descrito para o GEP foi utilizada uma arquitetura modular com camadas independentes. A figura 7.3 mostra a arquitetura utilizada. Na sequência, será descrita a funcionalidade de cada componente em sua respectiva camada da arquitetura.

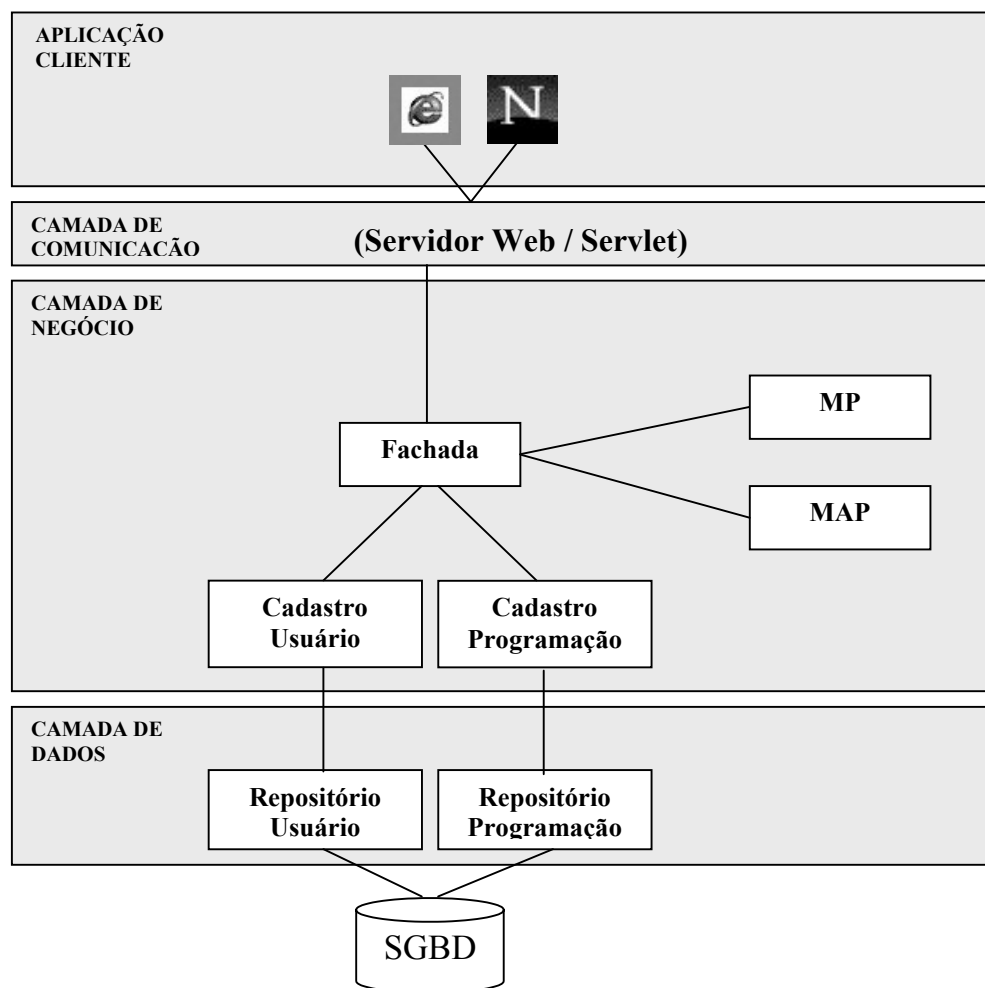


Figura 7.3 – Arquitetura do sistema GEP

7.4.1 Camada de Comunicação

A camada de comunicação é o ponto de acesso do usuário ao sistema. Nessa camada o servidor Web recebe as requisições do *browser* do usuário via protocolo HTTP. O servidor *Web* utilizado possui *servlet container*¹, onde é executado o *Servlet* que recebe as requisições de serviço do usuário e as encaminha para a camada de negócio. O *Servlet* é também responsável pela montagem da página de resposta que é encaminhada via protocolo HTTP ao usuário.

7.4.2 Camada de Negócio

A camada de negócios é o núcleo do sistema e é responsável pela implementação da lógica de negócios. A seguir são descritos os componentes dessa camada.

Fachada

Esse componente representa todos os serviços do sistema e funciona como servidora da camada de negócio. Ela possui referência para os cadastros do sistema, para os quais são requisitadas instâncias dos dados persistentes, e para as classes de negócio MP e MAP. Esse componente é implementado segundo o padrão *Singleton* [Gamma 94], garantindo que apenas uma instância estará ativa em algum momento durante a execução do sistema, evitando assim inconsistências. Quando a fachada recebe a solicitação de um serviço, ela coleta os dados necessários para execução do serviço através dos cadastros e os repassa para a classe de negócio responsável. A resposta do serviço da classe de negócio é repassada para o cliente. No caso de consultas comuns aos dados, como por exemplo a consulta de programação do dia, a fachada apenas repassa a requisição para o cadastro e retorna ao usuário a resposta deste.

¹ Módulo de suporte a execução de Servlet. No GEP o servidor Web usado foi o Jakarta Tomcat que possui módulo para execução de Servlet Java.

Cadastros

Os cadastros são responsáveis pela comunicação com a camada de dados. Os cadastros pertencem à camada de negócio porque encapsulam as verificações e validações inerentes ao negócio, como verificar se determinada instância já existe antes de inseri-la. Os cadastros do GEP são os seguintes:

Cadastro Usuário: Responsável pelo acesso aos dados referentes aos usuários, como cadastrar e recuperar usuários, registrar e recuperar preferências de configuração e avaliações.

Cadastro Programação: Responsável pelo acesso aos dados referentes à programação, como recuperar programas e programação.

Módulo de Personalização

Classe de negócio que implementa o serviço de personalização (Módulo de Personalização). Essa classe deve implementar a interface da figura 7.4.

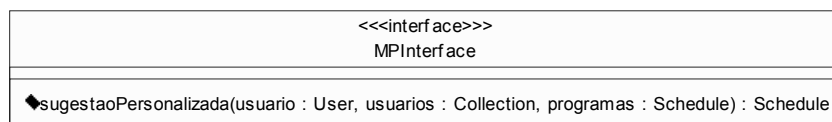


Figura 7.4 – Interface do Módulo de Personalização

A interface possui apenas o método *sugestaoPersonalizada()*, que recebe o usuário alvo (*User*), a coleção de usuários (*Collection*) do sistema e os programas que se deseja filtrar (*Schedule*). Esse método retorna uma programação contendo os programas de maior interesse do usuário segundo o seu perfil. Quando a recomendação é requisitada à fachada, essa é responsável por obter os parâmetros necessários junto aos cadastros e por chamar o método que gera um guia de recomendações de programas.

Para o protótipo do GEP, foi feita uma classe que implementa essa interface utilizando o algoritmo descrito no capítulo 5 desta dissertação. Entretanto, devido à implementação modular feita no sistema, outro algoritmo de filtragem poderia ser

utilizado, bastando para isso que a interface *MAPInterface* seja implementada, uma vez que a geração de guias personalizados depende unicamente dessa interface.

Módulo de Aquisição do Perfil

Classe de negócio que implementa o serviço de aprendizagem ativa do perfil do usuário (Módulo de Aquisição do Perfil). Essa classe deve implementar a interface da figura 7.5.

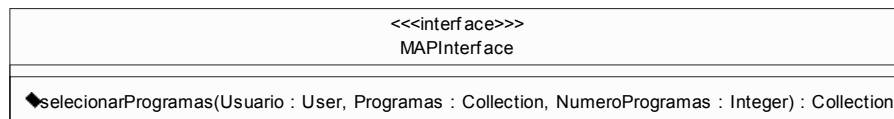


Figura 7.5 – Interface do Módulo de Aquisição de Perfil

A interface possui apenas o método *selecionarProgramas()* que recebe o usuário alvo (*User*), a coleção de programas (*Collection*) do sistema e o número de programas que serão retornados. Esse método retorna coleção de programas cuja avaliação ajuda a aprender mais rapidamente sobre o perfil do usuário. Quando uma seleção de programas para avaliação de um usuário é requisitada à fachada, essa é responsável por obter os parâmetros necessários junto aos cadastros para a chamada desse método

Para o protótipo do GEP foi feita uma classe que implementa essa interface utilizando o algoritmo descrito no capítulo 6 desta dissertação. Entretanto, devido à implementação modular feita no sistema, outros algoritmos de *selective sampling* podem ser usados, bastando para isso que a interface *MAPInterface* seja implementada, uma vez que a seleção de programas depende unicamente dessa interface.

7.4.3 Camada de Dados

A camada de dados é responsável pela manipulação da estrutura física de armazenamento de dados. Ela isola o sistema do meio de armazenamento de maneira que este pode ser trocado sem afetar as classes das camadas de negócio e de

comunicação. Cada cadastro da camada de negócio possui um repositório correspondente na camada de dados. Os repositórios que implementam o acesso aos dados no sistema GEP são as seguintes.

Repositório Usuário: Responsável por cadastrar e recuperar usuários, registrar e recuperar preferências de configuração e avaliações no SGBD. Recebe requisições de operações do Cadastro Usuário e monta o SQL que executa a operação requisitada no SGBD.

Repositório Programação: Responsável por recuperar os programas e programações no SGBD. Recebe requisições de operações do Cadastro Programação e monta o SQL que executa a operação requisitada no SGBD.

Para realizar operações dos repositórios no SGBD os cadastros possuem conexões JDBC com o SGBD.

7.5 Apresentação do GEP

Nessa seção a funcionalidade do GEP é ilustrada através de telas do sistema. O protótipo demonstrado foi carregado com programas na categoria de filmes. Os usuários do MovieLens [McJones 97] foram cadastrados como usuários do sistema e suas avaliações para os filmes inseridas no seu perfil para que fossem usados para a filtragem colaborativa. Um conjunto de canais fictícios foi criado e uma programação diária formada por filmes foi também criada para cada data.

Ao entrar no sistema o usuário vê uma tela de apresentação do sistema mostrada na figura 7.6. Nessa tela ele já tem a opção de navegar pelo conteúdo do sistema, podendo fazer consultas à programação do dia por local (canal de apresentação) e por gênero do programa.



Figura 7.6 – Tela inicial do sistema GEP

Quando executa uma consulta comum, o usuário utiliza o sistema como um usuário anônimo. A informação que ele obtém no sistema não pode ser filtrada porque ele não é cadastrado dentro do sistema, e conseqüentemente não existe qualquer conhecimento sobre seu perfil. A figura 7.7 demonstra o resultado de uma consulta à programação do canal intitulado HBO.



Figura 7.7 – Consulta à programação do canal HBO.

O usuário tem também a opção de conectar-se no sistema informando seu *login* e senha caso já seja cadastrado no sistema. Caso não seja, o usuário pode se cadastrar mediante o preenchimento de uma ficha cadastral onde são requisitadas informações gerais como nome, *login*, senha, e-mail¹ e telefone².

O cadastro no sistema é simples e a informação requisitada tem utilidade de identificação única do usuário dentro do sistema para que as avaliações que ele fornecer sejam associadas unicamente ao seu perfil e que possam ser utilizadas na geração de recomendações.

Quando o usuário está cadastrado no sistema, após conectar-se ao mesmo, a tela da figura 7.8 é apresentada.

¹ O e-mail do usuário é requisitado para que o sistema possa enviar guias de programação para o e-mail do usuário se solicitado. Esse serviço não está demonstrado neste trabalho.

² O número do celular do usuário é requisitado para que o sistema possa enviar recomendações de programas via SMS (*Short Message System*). Esse serviço ainda não está implementado no sistema.



Figura 7.8 – Tela inicial para o usuário conectado

Nessa tela o usuário tem opções de configuração do sistema, aprendizagem do perfil e requisição de sugestões.

Configuração

O usuário cadastrado pode configurar o sistema para melhor atender suas necessidades. As opções de configuração são as seguintes:

Preferências Gerais: o usuário configura os períodos (manhã, tarde e noite) da programação para os quais deseja receber recomendação, o número de sugestões por guia e a forma com que os itens são ordenados (por horário ou por canal).

Preferências de Canais: o usuário informa quais são os canais que deseja receber recomendações de programas.

Sugestão

O usuário do GEP pode requisitar uma sugestão de programação a qualquer instante. As sugestões do sistema são recomendações de programas que foram selecionados pelo algoritmo de filtragem como os mais similares ao perfil armazenado do usuário. As sugestões podem ser pedidas na programação da data corrente, da data seguinte e de toda a semana. O número de programas que aparecerão em uma sugestão é o número definido em Preferências Gerais (número de sugestões por guia), selecionados na ordem crescente de similaridade com o perfil do usuário do mais ao menos similar. A figura 7.9 ilustra uma tela com recomendações.

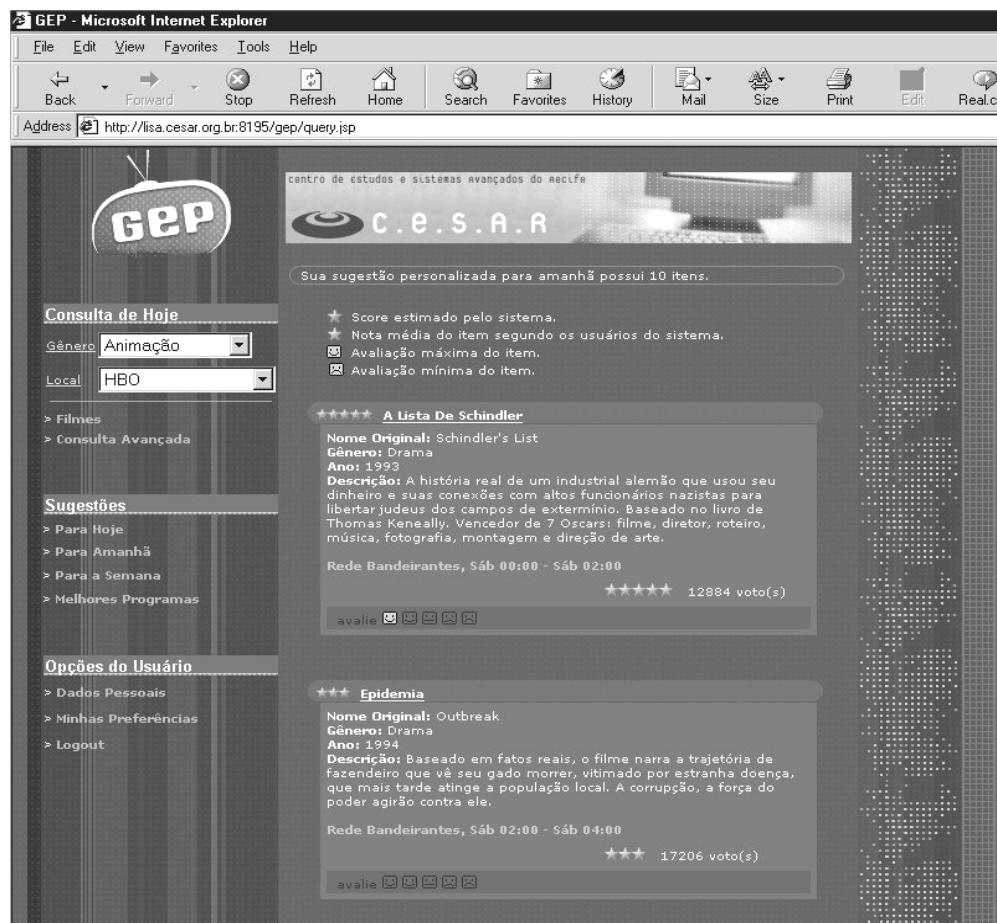


Figura 7.9 – Recomendação de programas para um usuário.

Quando o usuário está cadastrado no sistema ele poderá fornecer avaliações para qualquer programa que desejar. Para isso o GEP posiciona abaixo de cada programa mostrado ao usuário caixas de avaliação.

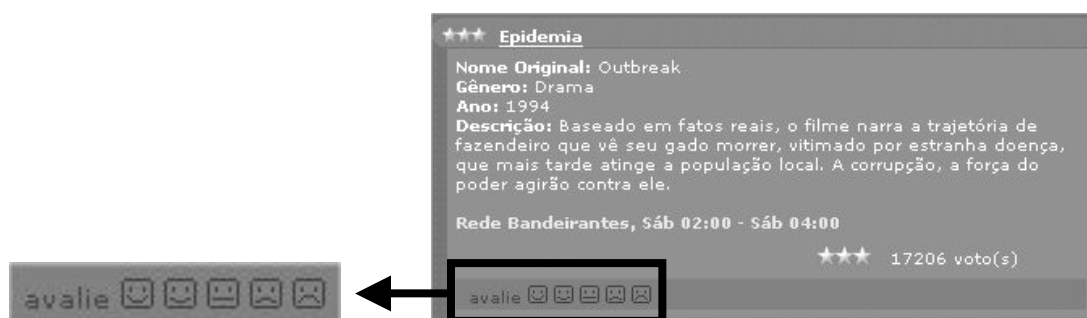


Figura 7.10- Caixas de avaliação do usuário. A feição do desenho indica o grau de satisfação do usuário com o programa de ótimo a péssimo.

A figura 7.10 mostra um programa com as caixas de avaliação. Quando o usuário clica em uma das caixas, o sistema registra essa avaliação no seu perfil com um escore que varia entre 1 (péssimo) e 5 (ótimo). Quando o usuário está cadastrado no sistema, qualquer consulta à programação ou sugestão do sistema apresentará os programas com essas caixas de avaliação.

Obtenção do Perfil

Para fornecer o seu perfil ao sistema, o usuário pode utilizar as seguintes funcionalidades do mesmo:

Preferências de Programas

Nessa opção o usuário pode remover e adicionar ou alterar as avaliações que ele fez no passado. A figura 7.11 mostra a tela dessa opção.

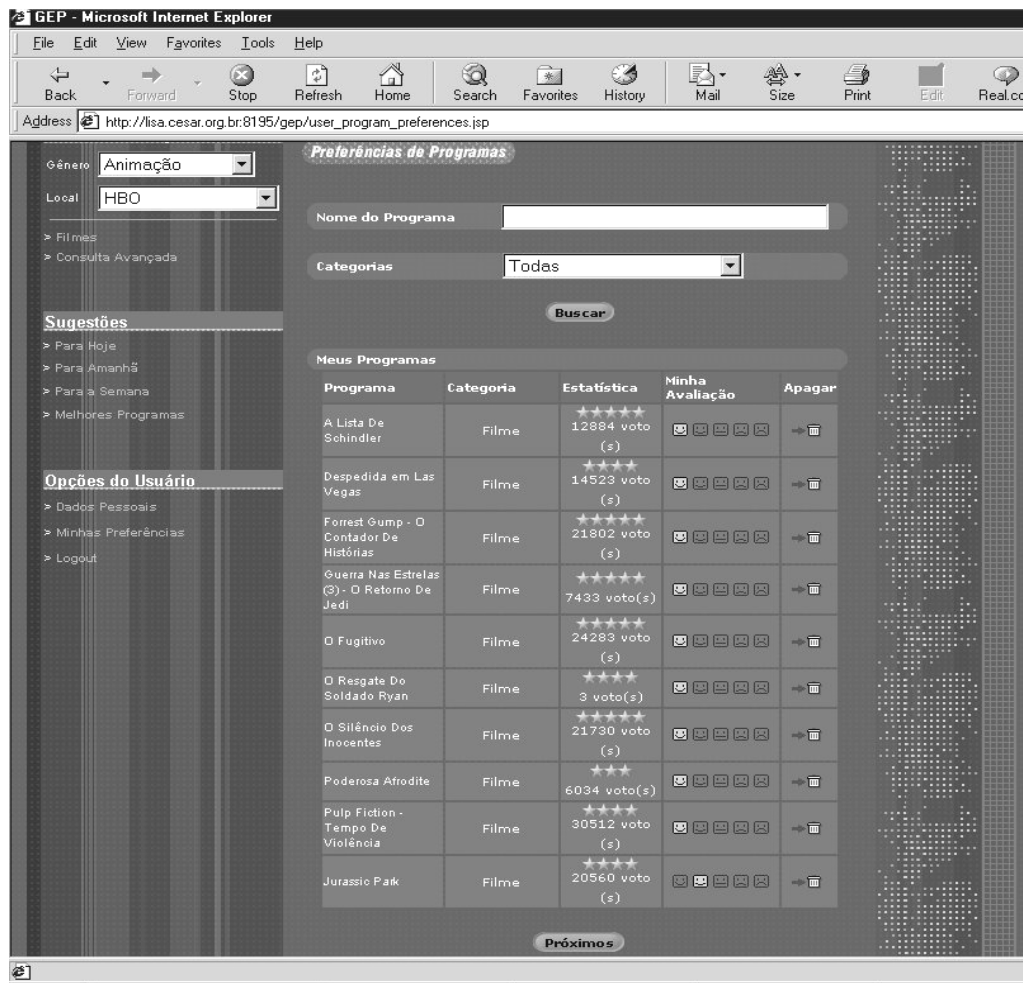


Figura 7.11 – Tela de preferências de programa.

Nessa tela os programas avaliados pelo usuário são listados com sua respectiva avaliação. O usuário pode remover a avaliação clicando sobre a lixeira correspondente ao programa, ou pode avaliar o programa de forma diferente. Ele poderá também fazer uma busca por outros programas para inserir no seu perfil.

Definir Meu Perfil

Nessa opção, o usuário tem a possibilidade de construir ou melhorar o seu perfil através da avaliação de programas indicados pelo próprio sistema. Será apresentada a ele uma lista de programas selecionados pelo sistema como mais relevantes, para que o sistema possa aprender sobre o seu perfil. Essa é a parte do sistema que executará o algoritmo de aprendizagem ativa implementado no MAP. A figura 7.12 demonstra a interface dessa opção.



Figura 7.12 – Tela de aprendizagem ativa do perfil do usuário.

Nessa tela o usuário recebe uma lista dos programas selecionados pelo sistema. Após avaliar os programas da lista corrente, o sistema seleciona novos programas e mostra uma nova lista ao usuário, que novamente avalia os itens de sua escolha. Esse processo é uma forma de aprendizagem iterativa, onde o sistema escolhe os programas apresentados ao usuário, de tal forma que a avaliação desses programas ajude a melhorar a qualidade das recomendações para esse usuário com uma quantidade pequena de avaliações.

7.6 Conclusão

Esse capítulo apresentou o GEP (Guia de Entretenimento Personalizado), um sistema de recomendação de entretenimento, desenvolvido seguindo uma arquitetura de camadas independentes, que permite que a comunicação com o usuário seja

modificada ou que o armazenamento de dados seja diferente sem alterar as outras camadas .

Esse sistema ilustra a aplicação dos algoritmos de filtragem de informação e de aprendizagem ativa do perfil do usuário discutidos ao longo dessa dissertação. A filtragem de informação do capítulo 5 foi aplicada na geração de recomendações do sistema, mais especificamente no módulo MP do sistema, enquanto a aprendizagem ativa do capítulo 6 foi aplicada na aquisição do perfil do usuário, mais especificamente no módulo MAP do sistema. Esses módulos foram planejados para serem transparente ao sistema quanto à forma com que são implementados, podendo assim serem facilmente substituídos por módulos que implementam outros algoritmos.

Por fim, o sistema e sua funcionalidade foram apresentados por meio de telas de demonstração de sua utilização. As principais funcionalidades demonstradas foram a geração de guias de recomendação (sugestões) e a aquisição do perfil do usuário através de avaliações dos programas apresentados em consultas e sugestões e através de um processo de aprendizagem ativa de perfil.

Capítulo 8

Conclusões

Este trabalho está relacionado com um problema que vem se tornando cada vez mais comum, o excesso de informação. O ser humano não conseguiu acompanhar a velocidade com que a informação chega até ele e agora procura soluções automatizadas para ajuda-lo a encontrar a informação apropriada.

Dentre as soluções existentes, a filtragem de informação vem se tornando cada vez mais popular, principalmente devido ao surgimento de sistemas de recomendação, capazes de personalizar o acesso do usuário à informação. Para executar tal tarefa, um sistema de recomendação deve ser capaz de identificar o perfil de cada usuário e, utilizando conhecimento sobre suas preferências, entregar a informação relevante para ele.

O foco desse trabalho foi a aquisição das preferências do usuário em um sistema de recomendação. O modelo de aprendizagem sobre as preferências dos usuários, geralmente feito através de avaliações do usuário sobre a informação, requer do mesmo um esforço necessário para avaliar a informação, o que tende a desinteressá-lo em usar o sistema [Nichols 97]. A alternativa de utilizar indicadores implícitos da preferência do usuário apresenta limites sobre o custo de obtenção, transporte e armazenamento desses indicadores além de refletir de maneira menos precisa a preferência dos usuários [Claypool 01].

Nesse trabalho foi desenvolvida uma metodologia original de aprendizagem ativa do perfil do usuário. O método criado para a seleção dos itens que o usuário avalia permite identificar o perfil dos usuários em um sistema de recomendação de forma mais eficiente do que no modelo usado pelos sistemas atuais.

8.1 Contribuições

O principal objetivo desta dissertação foi alcançado através do desenvolvimento de um método original de aprendizagem ativa do perfil dos usuários no algoritmo de FC com vizinhos mais próximos.

Para desenvolver o método *ActiveCP* foi feito primeiro uma reflexão sobre os atuais métodos de seleção de exemplos relevantes em algoritmos de aprendizagem

ativa e de redução de exemplos em KNN. Após verificar as diferenças entre esses métodos e o nosso problema, identificamos o conceito de ganho de informação e incerteza da classificação de um exemplo e traduzimos esses conceitos para o nosso problema. Introduzimos então a idéia do ganho de informação da avaliação de um item e sugerimos a utilização dos critérios da controvérsia e da popularidade para selecionar os itens cujas avaliações ajudariam a aprender mais rapidamente sobre o perfil do usuário. Fizemos também uma reflexão sobre cada um desses critérios para definir como medi-los e que parte das avaliações do sistema usar nessa medida.

Para alcançar o nosso segundo objetivo, realizamos experimentos para comprovar a eficiência do método criado. Para testar o desempenho dos critérios sugeridos na seleção dos itens, os experimentos foram organizados para realizar uma validação cruzada com os itens avaliados por cada usuário testado e para verificar a eficiência das predições feitas com o algoritmo de FC com vizinhos mais próximos usando as avaliações dos itens selecionados. A organização dos experimentos com validação cruzada não é praticada na área de sistemas de recomendação, podendo a metodologia criada para este trabalho ser usada como referência para trabalhos futuros na área.

Nos testes realizados, os critérios da controvérsia e da popularidade para a seleção dos itens apresentaram performance média melhor do que a seleção aleatória dos itens, porém a diferença na maior parte das vezes não foi estatisticamente significativa. Para melhorar os resultados criamos a metodologia *ActiveCP*, que combina os critérios da controvérsia e da popularidade na seleção dos itens que o usuário avalia. A combinação dos dois critérios mostrou resultados melhores do que a seleção utilizando apenas um dos critérios isoladamente e teve desempenho superior (inclusive estatisticamente significativo) ao da seleção aleatória dos itens, sendo alcançado nosso segundo objetivo de comprovar a superioridade do nosso método de aprendizagem ativa em relação ao método de aprendizagem do perfil do usuário nos sistemas atuais.

Além da criação de um método de aprendizagem ativa para a aquisição dos perfis dos usuários, a análise feita sobre a relevância da tarefa de um sistema de recomendação na escolha das métricas que serão usadas na sua avaliação e a sugestão das métricas ROC e Breese são contribuições relevantes para futuros trabalhos que também precisem avaliar um sistema de recomendação.

Como contribuições adicionais deste trabalho pode-se citar também o pioneirismo na pesquisa relacionada com filtragem de informação e sistemas de recomendação no CIn da UFPE. O trabalho motivou a pesquisa sobre o tema de sistemas de recomendação e filtragem de informação no centro, onde novas pesquisas poderão se beneficiar do trabalho realizado, diminuindo o esforço inicial.

O nosso último objetivo foi alcançado através do desenvolvimento do sistema GEP. Esse sistema ajudou a mostrar o potencial comercial da tecnologia de sistemas de recomendação e personalização no mercado nacional através da sua aplicação em sistemas informação ou de comércio eletrônico, sendo desenvolvido em uma instituição atuante no mercado de tecnologia da informação nacional como o CESAR (Centro de Estudos e Sistemas Avançados do Recife).

Este trabalho também ajudou a consolidar a parceria entre LIP 6 (*Laboratoire d'Informatique Paris 6*) e CIn-UFPE, e contou com a participação dos parceiros do LIP 6, em especial Vincent Corruble na idealização do problema e desenvolvimento da solução encontrada.

8.2 Trabalhos Futuros

Como continuação do trabalho desta dissertação sugere-se aprofundar os estudos feitos sobre os critérios da controvérsia e da popularidade na seleção dos itens que o usuário avalia.

Para medir a controvérsia, outras medidas de dispersão podem ser usadas além da variância, como a variância ponderada pela similaridade com o usuário alvo mostrada na seção 6.1.3. No problema da cobertura e intensidade da controvérsia citado, a abordagem usada foi de fixar o número de usuários usados para calcular a

controvérsia. Essa solução obteve êxito nos testes realizados na base de dados usada porque o número de avaliações para os itens é grande o suficiente. Entretanto, para sistemas com poucas avaliações por item, essa abordagem pode não ser apropriada pois é possível que não haja usuários suficientes que avaliaram um item para calcular a controvérsia. Uma sugestão de trabalho futuro seria utilizar uma metodologia para penalizar medida de controvérsia quando ela fosse calculada baseada em um número pequeno de avaliações.

No trabalho realizado, tanto a controvérsia quanto a popularidade dos itens foram calculadas globalmente, ou seja, no conjunto de todos os usuários do sistema que avaliaram o item. Intuitivamente os critérios da controvérsia e da popularidade de um item podem ser mais relevantes na sua seleção para formar o perfil de um usuário quando calculados entre os vizinhos desse usuário. Além disso, a utilização da vizinhança permite o uso de novas medidas que dependem da similaridade com o usuário alvo, como é o caso da variância cujos termos são ponderados pela similaridade dos usuários com o usuário alvo. A análise da utilização dos critérios localmente representa um problema bem mais complexo e, apesar de já termos iniciado testes com essa abordagem, os resultados ainda não são conclusivos, permanecendo portanto como sugestão de trabalho futuro.

Além da continuidade da aprendizagem ativa desenvolvida para o algoritmo de FC com vizinhos mais próximos, também é sugerido como trabalho futuro o desenvolvimento da aprendizagem ativa também para algoritmos de filtragem baseada em conteúdo. Atualmente a maioria dos sistemas utilizam a filtragem híbrida, que combina as filtragens colaborativa e baseada em conteúdo, o que torna importante que esses sistemas sejam capazes também de realizar a aprendizagem ativa na filtragem baseada em conteúdo.

Para o sistema GEP apresentado sugere-se a extensão do algoritmo de filtragem implementando um algoritmo híbrido, o que poderia melhorar a qualidade das suas recomendações além de evitar problema próprios da filtragem colaborativa.

Referências Bibliográficas

- [**Angluin 88**] Angluin, D. 1988. *Queries and concept learning*. Machine Learning, 6(1): p. 37-66.
- [**Aurenhammer 91**] Aurenhammer, F. 1991. *Voronoi diagrams – a survey of a fundamental geometrical data structure*. ACM Computing Surveys, 23: p. 345-405.
- [**Avery 99**] Avery, C., Resnick, P., Zeckhauser, R. 1999. *The market for Evaluations*. American Economic Review, 89(3): p.552-559.
- [**Balabanovic 97**] Balabanovic, M., Shoham, Y. 1997 *Fab: Content-Based, Collaborative Recommendation*. Communications of the ACM, 40(3): p. 66-72
- [**Baeza-Yates 99**] Baeza-Yates, R., Ribeiro-Neto, B. 1999. *Modern Information Retrieval*. 1ª Edição, Addison Wesley Longman.
- [**Belkin 92**] Belkin, N., Croft, B. W. 1992. *Information Filtering and Information Retrieval : Two Sides of the Same Coin ?* Communications of the ACM 35(12): p.29-38.
- [**Billsus 98**] Billsus, D., Pazzani, M. J. 1998 *Learning Collaborative Information Filters*. Fifteenth International Conference on Machine Learning, p. 46-54.
- [**Breese 98**] Breese, J. S., Heckerman, D., Kadie, C. 1998. *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, p. 43-52.
- [**Burke 99**] Burke, R. 1999. *The Wasabi Personal Shopper: A Case-Based Recommender System*. Proceedings of the 11th National Conference on Innovative Applications of Artificial Intelligence, p. 844-849.
- [**Claypool 99**] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., Sartin, M. 1999. *Combining Content-Based and Collaborative Filters in an Online Newspaper*. Proceedings of ACM SIGIR Workshop on Recommender Systems. Disponível em <http://www.cs.umbc.edu/~ian/sigir99-rec/>
- [**Claypool 01**] Claypool, M., Le, P., Waseda, M. e Brown, D. 2001. *Implicit Interest Indicators*. ACM IUI 01 Intelligent User Interfaces, p.33-40.
- [**Cohen 95**] Cohen, W. 1995. *Fast Effective Rule Induction*. Machine Learning: Proceedings of the Twelfth International Conference, p.115-123.

- [Cohen 98] Cohen, W. W., Basu, C., Hirsh, H. 1998. *Recommendation as Classification: Using Social and Content-Based Information in Recommendation*. Proceedings of the AAAI-98, p. 714 – 720.
- [Cohn 94] Cohn, D. A., Atlas, L., Lander, R. 1994. *Improving generalization with active learning*. Machine Learning, 15(2): p. 201-221.
- [Cotter 00] Cotter, P, & Smyth, B. 2000. *PTV: Intelligent Personalised TV Guides*. Proceedings of IAAI 2000 conference, p. 957-964.
- [Gamma 94] Erich Gamma et al. 1994. *Design Patterns: Elements of Reusable Object Oriented Software*. 1ª Edição, Addison-Wesley.
- [Goldberg 92] Goldberg, D., Nichols, D., Oki, B. M. e Terry, D. 1992. *Using Collaborative Filtering to Weave an Information Tapestry*. Communications of the ACM, 35(12): p. 61-70.
- [Good 99] 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*. Proceedings of IAAI 99 conference, p. 439-446.
- [Hanley 82] Hanley, J. A., McNeil, B. J., 1982. *The Meaning and use of the Area Under a Receiver Operating Characteristic (ROC) Curve*. Radiology, 143: p. 29-36.
- [Hasenjager 98] Hasenjager, M., Ritter, H. 1998. *Active Learning with Local Models*. Neural Processing Letters, 7(2): p.107-117.
- [Herlocker 99] Herlocker, J. L., Konstan, J. A., Borchers, A., Riedl, J. 1999. *An Algorithmic framework for performing collaborative filtering*. Proceedings of the SGIR 99, p. 230-237.
- [Herlocker 00] Herlocker, J. L. 2000. *Understanding and Improving Automated Collaborative Filtering Systems*. Phd thesis, University of Minnesota.
- [Hill 92] Hill, W. C., Hollan, J. D., Wrobelwski, D. , McCandless, T. 1992. *Read Wear and Edit Wear*. Proceedings of ACM CHI 92, p. 3-9.
- [Hill 95] Hill, W., Stead, L., Rosenstein, M., Furnas, G. W. 1995. *Recommending and Evaluating Choices in a Virtual Community of Use*. Proceedings of ACM CHI 95, p. 194-201.
- [JSP 02] JSP: JavaServer Pages™. Capturado em abril de 2002. <http://java.sun.com/products/jsp>

- [Konstan 97] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., Riedl, J. 1997. *GroupLens: Applying Collaborative Filtering to Usenet News*. Communications of the ACM, 40(3). p. 77-87.
- [Krukwich 96] Krukwich, B., Burkey, C. 1996. *Learning user information interests through extraction of semantically significant phrases*. Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access.
- [Lewis 94] Lewis, D. D., Catlett, J. 1994. *Heterogeneous uncertainty sampling for supervised learning*. Proceeding of ICML 94, p. 148-156.
- [Lin 00] Lin, W., Alvarez, S. A., Ruiz, C. 2000. *Collaborative Recommendation via Adaptive Association Rule mining*. ACM-SIGKDD 2000 Workshop on Web Mining for E-Commerce. Disponível em <http://robotics.stanford.edu/~ronnyk/WEBKDD2000/papers/>
- [Lindenbaum 99] Lindenbaum, M., Markovitch, S., Rusakov, D. 1999. *Selective Sampling for Nearest Neighbor Classifiers*. Proceedings of The Sixteenth National Conference on Artificial Intelligence AAAI-99, p. 366-371.
- [Malone 87] Malone, T. W., Grant, K. R., Turbak, F. A., Brobst, S. A., Cohen, M. D. 1987. *Intelligent Information-Sharing Systems*. Communications of the ACM, 30(5): p. 390-402.
- [McJones 97] McJones, P. 1997. Eachmovie collaborative filtering data set. <http://www.research.digital.com/SRC/eachmovie>. DEC System Research Center.
- [Miller 96] Miller, B. N., Riedl, J. T., Konstan, J. A. 1996. *Experiences with GroupLens: Making Usenet Useful Again*. Proceedings of the USENIX 96 Annual Technical Conference, p. 219-233.
- [Mitchel 97] Mitchell, T. 1997. *Machine Learning*. 1ª Edição, McGraw Hill.
- [Morita 94] Morita, M., Shinoda, Y. 1994. *Information Filtering Based on User Behaviour Analysis and Best Match Text Retrieval*. Proceedings of ACM-SIGIR 94, p. 272-281.
- [Montgomery 97] Montgomery, D. C. 1997. *Design and analysis of experiments*. 4ª Edição, John Wiley & Sons.
- [Nichols 97] Nichols, D. M. 1997. *Implicit Ratings and Filtering*. 5th DELOS Workshop on Filtering and Collaborative Filtering. p. 10-12.
- [Oard 98] Oard, D., Kim, J. 1998. *Implicit FeedBack for Recommender Systems*. Proceedings of the AAAI WorkShop on Recommender Systems. Disponível em <http://www.glue.umd.edu/~oard/research.html>

- [**Page 98**] Page, L., Brin, S., Motwani, R., Winograd, T. 1998. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report, Department of Computer Science, Stanford University, CA. Disponível em www-db.stanford.edu/~backrub/pageranksub.ps.
- [**Pazzani 97**] Pazzani, M. and Billsus, D. 1997. *Learning and revising user profiles: The identification of interesting web sites*. Machine Learning 27(3): p. 313-331.
- [**Perny 02**] Perny, P., Zucker, J. D. 2002. *Preference-based Search and Machine Learning for Collaborative Filtering: the "Filme-Conseil" Movie Recommender System*. 13 Journal in Information Engineering Sciences, 1(1). Disponível em <http://www.revue-i3.org/volume01/numero01/index.htm>
- [**Resnick 94**] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J. 1994. *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work, p. 175-186.
- [**Rucker 97**] Rucker, J. e Polanco, M. J. 1997. *Personalized Navigation for the Web*. Communications of the ACM, 40(3): p. 73-89.
- [**Servlet 02**] Servlet: Java™ Servlet Technology. Capturado em abril de 2002. <http://java.sun.com/products/servlet/index.html>
- [**Shardanand 95**] Shardanand, U., Maes, P. 1995. *Social Information Filtering: Algorithms for Automating "Word of Mouth"*. Proceedings of CHI 95, p. 210-217.
- [**Swets 63**] Swets, J. A., 1963. *Information Retrieval Systems*. Science, 141: p. 245-250.
- [**Terveen 97**] Terveen, L., Hill, W., Amento, B., McDonald, D., Creter, J. 1997 *Phoaks: A System For Sharing Recommendations*. Communications of the ACM, 40(3): p.59-62.
- [**Terveen 01**] Terveen, L., Hill, W. 2001. *Beyond Recommender Systems: Helping People Help Each Other*. Capítulo 22 de Human-Computer Interaction in the Millenium, Addison-Wesley, 2001.
- [**Tong 00**] Tong, S., & Koller, D. 2000. *Support vector machine active learning with applications to text classification*. Proceedings of the Seventeenth International Conference on Machine Learning, p. 999-1006.
- [**Yao 95**] Yao, Y. Y., 1995. *Measuring Retrieval Effectiveness Based on User Preference of Documents*. Journal of the American Society for Information Science, 46(2): p. 133-145.
- [**Wilson 00**] Wilson, D. R., Martinez T. R. 2000. *Reduction techniques for exemplar-based learning algorithms*. Machine Learning, 38(3): p. 257-268.