

Sistemas de Recomendação

Eliseo Berni Reategui e Sílvia César Cazella

Abstract

Recommending products, items of information to an Internet user is one of the biggest challenges of the virtual world. The appropriate recommendation of a book, for example, can make the difference between getting or losing a customer. Because of this need to "conquer" the users, personalization has become a very important factor. This chapter gives an overview of recommender systems used in personalization tasks. It describes the most popular recommendation techniques, details the systems architecture and brings examples of their use both in the industry and academia. The chapter also discusses the privacy problem and presents future trends.

Resumo

Recomendar produtos, itens ou informações para um usuário da Internet apresenta-se como um dos maiores desafios no mundo virtual. A recomendação adequada de um livro, por exemplo, pode fazer a diferença entre conquistar o usuário ou perdê-lo. Devido a esta necessidade de conquista, a personalização tem se apresentado como um fator facilitador no momento de "cativar" o usuário. Este capítulo dá uma visão geral sobre os sistemas de recomendação utilizados nas tarefas de personalização. Descreve as técnicas e estratégias de recomendação mais utilizadas, detalha a arquitetura dos sistemas e traz exemplos de utilização destes tanto no meio acadêmico quanto na indústria. Discute ainda o problema da privacidade e aborda tópicos que apontam para novas tendências na área.

1. Introdução aos Sistemas de recomendação

Com a quantidade de informações e com a disponibilidade facilitada das mesmas pelo uso da Internet, as pessoas se deparam com uma diversidade muito grande de opções. Muitas vezes um indivíduo possui muito pouca ou quase nenhuma experiência pessoal para realizar escolhas entre as várias alternativas que lhe são apresentadas. A questão relevante neste momento refere-se a como proceder nestes casos? Para minimizar as

dúvidas e necessidades que temos frente à escolha entre alternativas, geralmente confiamos nas recomendações que são passadas por outras pessoas, as quais podem chegar de forma direta (*word of mouth*) [Maes and Shardanand 1995], cartas de recomendação, opiniões de revisores de filmes e livros, impressos de jornais, entre outros.

Os sistemas de recomendação auxiliam no aumento da capacidade e eficácia deste processo de indicação já bastante conhecida na relação social entre seres humanos [Resnick e Varian 1997]. Em um sistema típico as pessoas fornecem recomendações como entradas e o sistema agrega e direciona para os indivíduos considerados potenciais interessados neste tipo de recomendação. Um dos grandes desafios deste tipo de sistema é realizar o casamento correto entre os que estão recomendando e aqueles que estão recebendo a recomendação, ou seja, definir e descobrir este relacionamento de interesses.

Os proponentes do primeiro sistema de recomendação denominado Tapestry [Goldberg et al. 1992][Resnick e Varian 1997], criaram a expressão “filtragem colaborativa”, visando designar um tipo de sistema específico no qual a filtragem de informação era realizada com o auxílio humano, ou seja, pela colaboração entre os grupos de interessados. Os autores preferem utilizar a expressão sistemas de recomendação, por ser um termo genérico e defendem este posicionamento por dois motivos: primeiro porque os recomendadores podem não explicitar colaboração com os que as recebem, pois um pode não conhecer o outro, e por último os recomendadores podem sugerir itens de interesse particular, incluindo aqueles que poderiam ser desconsiderados. A questão aqui é como verificar o real interesse em colaborar, pois na realidade as recomendações esboçadas pelos recomendadores estão concentradas em seus interesses próprios. Na verdade o grupo a ser formado para receber as recomendações deveria pelo menos apresentar indivíduos com interesses comuns (explícitos) ou comportamento comum (implícito). Neste trabalho assume-se que sistemas de filtragem colaborativa e sistemas de filtragem baseada em conteúdo são tipos de sistemas de recomendação que aplicam abordagens distintas, mas possuem como finalidade única a recomendação.

Alguns autores, como Montaner [Montaner et al. 2003], destacam que existe um terceiro tipo de filtragem de informação denominada de filtragem demográfica. A filtragem demográfica utiliza a descrição de um indivíduo para aprender o relacionamento entre um item em particular e o tipo de indivíduo que poderia vir a se interessar. Este tipo de abordagem utiliza as descrições das pessoas para conseguir aprender o relacionamento entre um item e o tipo de pessoa que gostaria deste. O perfil do usuário é criado pela classificação dos usuários em estereótipos que representam as características de uma classe de usuários. Dados pessoais são requisitados ao usuário, geralmente em formulários de registro, e usados como caracterização dos usuários e seus interesses. Como exemplo, Montaner cita o método implantado por Krulwich (Krulwich apud [Montaner et al. 2003]) em LifeStyle Finder onde é utilizado um sistema demográfico denominado PRIZM da Claritas Corporation. Este sistema tem o objetivo de dividir a população americana em 62 agrupamentos demográficos de acordo com seus históricos de compra, características referentes ao tipo de vida e respostas a pesquisas.

Os websites de comércio eletrônico são atualmente o maior foco de utilização dos sistemas de recomendação, empregando diferentes técnicas para encontrar os produtos mais adequados para seus clientes e aumentar deste modo sua lucratividade. Introduzido em julho de 1996 o My Yahoo foi o primeiro web site a utilizar os sistemas de recomendação em grandes proporções, utilizando a estratégia de customização [Manber et al. 2000] (ver seção 3). Hoje em dia, um grande número de websites emprega os sistemas de recomendação para levar aos usuários diferentes tipos de sugestões, como ofertas casadas ("clientes que compraram item X também compraram item Y"), itens de sua preferência, itens mais vendidos nas suas categorias favoritas, etc.

Este capítulo apresenta os sistemas de recomendação, descreve as técnicas mais utilizadas e mostra como podem ser aplicados em exemplos práticos. Iniciaremos pelos mecanismos de coleta de informações geralmente utilizados (Seção 2), apresentando a seguir, na Seção 3, questões relativas a estratégias, e na Seção 4 questões relativas às técnicas. Na Seção 5 e 6, serão discutidas questões relativas a arquitetura de Sistemas de Recomendações e privacidade, respectivamente. No final do capítulo é apresentada uma conclusão que visa destacar a relevância do tema Sistemas de Recomendação.

2. Coleta de Informações

Para que se possa recomendar itens a um usuário, é necessário se ter conhecimento sobre quem é este usuário. É necessário capturar e armazenar seus dados pessoais e comportamentais relativos. Portanto, é essencial que se possa identificar o usuário no momento em que ele acessa o sistema onde foram implantadas as rotinas de recomendação.

Atualmente a identificação dos usuários é bastante comum no mundo dos negócios. Ao abrir uma conta bancária, adquirir um imóvel ou fazer compras pela internet o cliente é obrigado fornecer diversos dados, desde dados pessoais (nome, endereço, etc.), até números de documentos. Duas das formas mais habituais de identificação de usuário são:

Identificação no servidor: normalmente disponibiliza ao usuário uma área de cadastro com informações pessoais, tais como: nome, data de nascimento, sexo, endereço e outros. Além disso, solicita obrigatoriamente um login e senha. Estas informações ficam armazenadas em um banco de dados no servidor. Sempre que o usuário acessar o sistema, ele poderá fazer sua identificação/autenticação informando seu login anteriormente cadastrado. Este mecanismo permite que o web site identifique com mais precisão o usuário que nele se conecta.

Identificação no cliente: utiliza normalmente cookies, um mecanismo pelo qual um website consegue identificar que determinado computador está se conectando mais uma vez a ele. Este método assume que a máquina conectada é utilizada sempre pela mesma pessoa. Logo, ao identificar a máquina, o website está na realidade identificando seu usuário. Trata-se de um mecanismo mais simples do que a identificação através do servidor porém menos confiável, principalmente se o computador identificado for utilizado por mais de uma pessoa.

Depois de identificado o usuário, é possível coletar dados sobre este de forma implícita ou explícita. Na modalidade de coleta explícita (também conhecida como

customização), o usuário indica espontaneamente o que lhe é importante. Por exemplo, o usuário pode indicar que suas seções favoritas em uma livraria virtual são Design e Música, como ilustra a figura 2.1.

Figura 2.1. Seleção de seções favoritas

Estas informações ficam então armazenadas em uma tabela, como ilustrado a seguir.

Tabela 2.1. Dados sobre seções favoritas

Usuário	Seção favorita
paulo@livros.com.br	Música
paulo@livros.com.br	Design

Na modalidade implícita, através de ações do usuário infere-se informações sobre suas necessidades e preferências. Por exemplo, armazenando-se dados de navegação do usuário (páginas consultadas, produtos visualizados, etc) é possível detectar que ele se interessa por Design e Turismo. A tabela 2.2 mostra o tipo de informação que pode ser armazenada como consequência do acesso às páginas descrevendo os livros A001, B023 e D005.

Tabela 2.2. Dados cadastrais de clientes de uma empresa.

Usuário	Item	Categoria	Data Acesso
paulo@livros.com.br	A001	Design	5/4/2005
fabio@livros.com.br	B023	Design	5/4/2005
fabio@livros.com.br	D005	Turismo	8/4/2005

Através desta técnica, é possível conhecer melhor a preferência dos usuários sem que eles tenham que fornecer informações explicitamente, e em seguida utilizar estes dados para fazer recomendações. A próxima seção descreve as principais estratégias de recomendação e como estas são empregadas.

3. Estratégias de Recomendação

Os principais objetivos dos sistemas de recomendação são a fidelidade e o conseqüente aumento da lucratividade das empresas. Diferentes estratégias podem ser para personalizar ofertas para um usuário, cada uma exigindo um grau de complexidade distinto no tratamento de informações coletadas. As estratégias mais utilizadas são descritas nas subseções a seguir.

3.1. Listas de recomendação

Esta estratégia consiste em manter listas de itens organizados por tipos de interesses. Não há necessidade de análise mais profunda de dados do usuário para criação destas listas, apenas a observação dos tipos de itens mais populares, e ordenação destes em grupos tais como: "Itens mais vendidos", "Idéias para presentes", entre outros. A figura 3.1 apresenta um exemplo de lista de recomendações da loja Amazon.com, a maior livraria virtual do planeta. No exemplo é apresentada uma lista de presentes incluindo itens de música e cozinha. Na porção direita da tela são apresentadas outras listas, todas relacionadas a presentes para bebês.

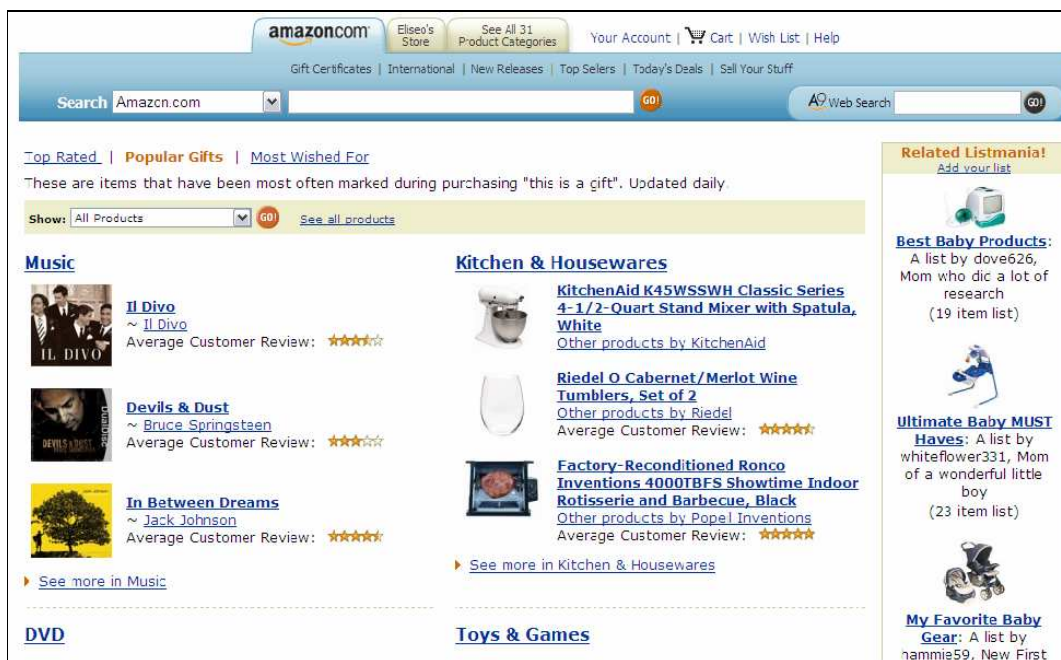


FIGURA 3.1. Lista de recomendações de presentes

A principal vantagem neste tipo de estratégia está na facilidade de implementação. Basta manter-se listas de acordo com as necessidades de marketing, de aumento de lucratividade, de disseminação de informações (no caso de recomendação de conteúdo), etc. A desvantagem é que as recomendações não são dirigidas a cada usuário independentemente, mas sim a todos os usuários sem distinção.

3.2. Avaliações de usuários

Uma das estratégias mais utilizadas em sistemas de recomendação são as avaliações dos usuários. Ou seja, além de comprar um produto o usuário também deixa um comentário sobre o item adquirido. É muito comum vermos opinião de usuários na forma de ícones dispostos ao lado do item visualizado, como ilustrado nas figuras a seguir.

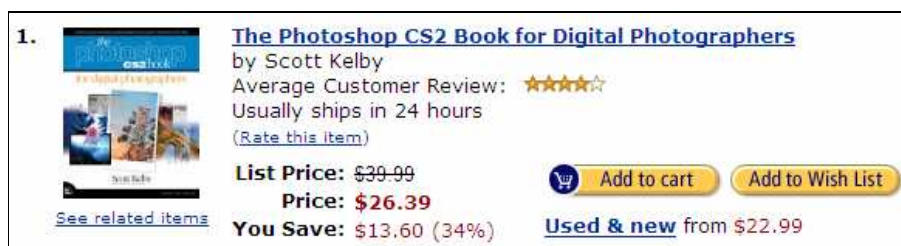


FIGURA 3.2. Avaliações de usuários (estrelas)



FIGURA 3.3. Avaliações de usuários (submarinos)

Avaliações de clientes são muito úteis para assegurar outros consumidores da qualidade e utilidade dos produtos comercializados. No entanto, para que um sistema possa funcionar corretamente com base nos comentários do usuário, é preciso que haja veracidade das opiniões fornecidas. Alguns sistemas propõem mecanismos para incentivar os usuários a contribuir com opiniões verídicas, através de brindes, bônus, etc.

Do ponto de vista de implementação, este também é um mecanismo fácil de implementar, na medida em que não exige nenhum tipo de dispositivo inteligente. Basta armazenar e disponibilizar as avaliações de usuários sobre os itens tratados, e apresentar estas opiniões no momento apropriado.

3.3. Suas Recomendações

Este tipo de recomendação é oferecido em uma seção inteiramente dedicada a sugestões feitas especificamente para o usuário. Dois tipos de recomendação são possíveis nestas seções: aquelas feitas a partir de preferências implícitas ou explícitas. A figura a seguir apresenta um exemplo de página de recomendação onde são trazidas sugestões a partir de dados coletados de maneira implícita.




FIGURA 3.4. Página de recomendações

No exemplo apresentado, há uma lista de itens recomendados, entre livros e cds. Na figura, vemos dois deles (um livro de mineração de dados e um CD do músico Caetano Veloso). O sistema salienta, ao alto da página, que as recomendações feitas foram baseadas em 13 elementos adquiridos pelo usuário. Ou seja, o usuário não disse explicitamente que se interessava por CDs de música brasileira ou por livros de mineração de dados, mas estas informações foram inferidas a partir de seus dados de compra.

3.4. "Usuários que se interessaram por X também se interessaram por Y"

Este tipo de recomendação é obtido através de técnicas capazes de encontrar em uma base de dados associações entre itens avaliados por usuários (comprados, lidos,...). É outro tipo de recomendação muito comum em websites de comércio eletrônico, como mostra a figura a seguir.

Making It Personal: How to Profit from Personalization without Invading Privacy
 by Don Peppers (Foreword), Martha Rogers (Foreword), Bruce Kasanoff "Someday soon, when you read a book like this on your digital book reader, a record of your reading habits will be created in a..." (more)



Availability: Usually ships within 1-2 business days from these sellers.

31 used & new from \$0.88
Edition: Hardcover

[See all buying options](#)
[Add to Wish List](#)

[Search inside this book](#)

Customers who bought this book also bought

- The One to One Fieldbook (One to One) by DON PEPPERS
- Enterprise One to One by DON PEPPERS
- The One to One Future (One to One) by DON PEPPERS
- Jack Trout on Strategy by Jack Trout
- The Culting of Brands: When Customers Become True Believers by Douglas Atkin

FIGURA 3.5. Recomendação por associação

No exemplo ilustrado na figura 3.5, a partir da seleção de um livro de personalização outros livros similares são apresentados. Este tipo de recomendação, chamado aqui de recomendação por associação, é a forma mais complexa de recomendação. Ela exige uma análise mais profunda dos hábitos do usuário para a identificação de padrões e recomendação de itens com base nestes padrões. As seções 4.1.2 e 4.2 mostram como se pode chegar a estas recomendações a partir dos dados coletados.

3.5. Associação por conteúdo

Também é possível fazer recomendações com base no conteúdo de determinado item, por exemplo um autor, um compositor, um editor, etc. Para possibilitar este tipo de recomendação, é necessário que se encontrem associações num escopo mais restrito. Por exemplo: Os livros A e B de determinado autor são frequentemente vendidos em conjunto. A figura abaixo mostra este tipo de recomendação em uma livraria virtual.

Word of Mouse: The Market Power of Collaborative Filtering

by [John Riedl](#), [Joseph A. Konstan](#), [Eric Vrooman](#)



List Price: \$40.95
Price: **\$8.76**
You Save: \$2.19 (20%)
Availability: Available for download now

READY TO BUY?

 **Add to Shopping Cart**

Customers who bought titles by John Riedl also bought titles by this author:

- [Soumen Chakrabarti](#)

► [Explore similar authors](#)

FIGURA 3.6. Recomendação por associação

No exemplo, o sistema recomenda o autor Soumen Chakrabarti para um usuário que selecionou uma obra do autor John Riedl. Diferentes técnicas podem ser utilizadas para se obter este tipo de recomendação, apresentadas no próximo capítulo.

4. Técnicas

Várias tecnologias têm surgido visando a identificação de padrões de comportamento (consumo, pesquisa, etc.) e utilização destes padrões na personalização do relacionamento com os usuários. Estas técnicas fundamentam o funcionamento dos sistemas de recomendação e são apresentadas nas subseções a seguir.

4.1. Filtragem de informações

A demanda por tecnologias de filtragem de informação não é algo novo [Loeb e Terry 1992]. Peter Denning já escrevia em um volume da *Communications of ACM* do ano de 1982 (Peter Denning apud [Loeb e Terry 1992]) sobre a preocupação no que se refere à quantidade de informação que estava sendo gerada pelos diversos tipos de sistema e recebidas pelos usuários. Ele destacava que toda a atenção estava concentrada na geração da informação para suprir as necessidades do usuário. Destacava também que era importante se preocupar com o recebimento da informação, com o controle de processo, de recuperação e filtragem da informação para que esta alcançasse a pessoa que deveria utilizá-la.

A questão portanto é: Como deve ser entendida a expressão filtragem de informação? Para Belvin [Belvin e Croft 1992] filtragem de informação é o nome utilizado para descrever uma variedade de processos que envolvem a entrega de informação para as pessoas que realmente necessitam delas. Com o tempo este termo tornou-se muito popular e artigos técnicos foram escritos descrevendo o emprego de filtragem em diversas aplicações, como e-mail, documentos eletrônicos em escritórios, entre outros. No entanto, a distinção entre filtragem e processos relacionados com recuperação de informações não era sempre clara. As expressões “recuperação de informação” e “filtragem de informação” descrevem soluções que procuram auxiliar na

solução de problemas referentes a sobrecarga de informações. A expressão “recuperação de informação” tradicionalmente envolve armazenamento, índices, e tecnologia para recuperação de documentos textuais. Para que a recuperação realmente ocorra o usuário deve descrever sua necessidade de informação em um formulário, com a consulta (*query*) do que necessita. A partir deste momento, o sistema busca casar a necessidade explicitada com os documentos armazenados. Este tipo de abordagem tende a manter uma base com característica mais estática no armazenamento das informações, e a início da interação é provocada pelo usuário.

Diferentemente da recuperação [Foltz e Dumais 1992] a filtragem propõe uma abordagem distinta. Esta abordagem geralmente mantém um perfil dos interesses do usuário. A idéia é que esta abordagem tenha como característica um maior tempo de duração no que tange ao casamento de interesses. Ou seja, a filtragem não se refere ao momento, mas sim a preferências, enquanto a recuperação baseia-se na percepção de uma necessidade do momento. Esta filtragem deve ser aplicada a cada novo item adicionado procurando verificar se este atende ao usuário. A seguir são descritas técnicas de filtragem aplicáveis a sistemas de recomendação.

4.1.1. Filtragem Baseada em Conteúdo

Segundo Herlocker [Herlocker 2000], por muitos anos os cientistas têm direcionado seus esforços para aliviar o problema ocasionado com a sobrecarga de informações através de projetos que integram tecnologias que automaticamente reconhecem e categorizam as informações. Alguns *softwares* têm como objetivo gerar de forma automática descrições dos conteúdos dos itens e comparar estas descrições com os interesses dos usuários visando verificar se o item é ou não relevante para cada um [Balabanovic e Shoham 1997]. Esta técnica é chamada de filtragem baseada em conteúdo [Herlocker 2000][Ansari et al. 2000] por realizar uma seleção baseada na análise de conteúdo do itens e no perfil do usuário.

A descrição de interesses do usuário é obtida através de informações fornecidas por ele próprio ou através de ações, como seleção e aquisição de itens. Muitas ferramentas que esta abordagem aplicam técnicas como indexação de frequência de termos (Salton apud [Herlocker 2000]). Neste tipo de indexação, informações dos documentos e necessidades dos usuários são descritas por vetores com uma dimensão para cada palavra que ocorre na base de dados. Cada componente do vetor é a frequência que uma respectiva palavra ocorre em um documento ou na consulta do usuário. Claramente, os vetores dos documentos que estão próximos aos vetores da consulta do usuário são considerados os mais relevantes para ele.

Outros exemplos de tecnologias aplicadas para filtragem baseada em conteúdo são índices de busca booleana, onde a consulta constitui-se em um conjunto de palavras-chave unidas por operadores booleanos (Cleverdon apud [Herlocker 2000]); sistemas de filtragem probabilística, onde raciocínio probabilístico é aplicado para determinar a probabilidade que um documento possui de atender as necessidades de informação de um usuário (Fuhr, Robertson, Wong apud [Herlocker 2000]); e interfaces de consultas com linguagem natural, onde segundo o autor as consultas são colocadas em sentenças naturais (Jacobs, Lewis, Strzalkowski apud [Herlocker 2000]).

Uma maneira de trabalhar com a filtragem baseada em conteúdo é através de uma solicitação de análise de itens feita ao próprio usuário, onde este deve avaliar alguns itens indicando se estes são de interesse ou não. Uma vez realizada a avaliação, o sistema busca itens que "casam" em conteúdo com o que foi classificado como de interesse, e desconsidera os que "casam" em conteúdo como o que foi classificado de não interesse.

Sistemas deste tipo apresentam limitações como: o conteúdo de dados pouco estruturados é difícil de ser analisado (e.g. vídeo e som); o entendimento do conteúdo do texto pode ser prejudicado devido a uso de sinônimos; pode ocorrer a super especialização, pois o sistema procura se basear em avaliações positivas e negativas feitas pelo usuário, não apresentando conteúdos que não fechem com o perfil.

4.1.2. Filtragem Colaborativa

A abordagem da filtragem colaborativa foi desenvolvida para atender pontos que estavam em aberto na filtragem baseada em conteúdo [Herlocker 2000][Ansari et al. 2000]. A Filtragem Colaborativa se diferencia da filtragem baseada em conteúdo exatamente por não exigir a compreensão ou reconhecimento do conteúdo dos itens.

Nos sistemas colaborativos a essência está na troca de experiências entre as pessoas que possuem interesses comuns. Nestes sistemas, os itens são filtrados baseado nas avaliações feitas pelos usuários. Por exemplo, o primeiro sistema criado com esta abordagem foi o Tapestry [Goldberg et al. 1992] que permitia ao usuário especificar um consulta como: “mostre-me todos os memorandos que uma determinada pessoa considera como importante”. Membros de determinada comunidade podiam deste modo ser beneficiados pela experiência de outros.

Segundo Herlocker [Herlocker 2000] os primeiros sistemas de filtragem colaborativa requeriam usuários para especificar o relacionamento de predição entre suas opiniões, ou de modo explícito indicar os itens de interesse. Porém, em seguida estes sistemas automatizaram todo o procedimento através da coleção das pontuações dos itens pelos usuários. Um usuário de um sistema de filtragem colaborativa deve portanto pontuar cada item experimentado, indicando o quanto este item casa com sua necessidade de informação. Estas pontuações são coletadas para grupos de pessoas, permitindo que cada usuário se beneficie das pontuações (experiências) apresentadas por outros usuários na comunidade.

Sistemas de filtragem colaborativa simples apresentam para o usuário uma média de pontuações para cada item com potencial de interesse. Esta pontuação permite ao usuário descobrir itens que são considerados de interesse pelo grupo e evitar os itens que são considerados de pouco interesse. Sistemas mais avançados descobrem de maneira automática relações entre usuários (vizinhos mais próximos), baseado na descoberta de padrões comuns de comportamento. Um exemplo de ambiente baseado em filtragem colaborativa é o sistema de recomendação de filmes MovieLens [Riedl et al. 1999]. Nele o usuário insere pontuações para filmes que tenha visto e o sistema utiliza estas pontuações para encontrar pessoas com gostos similares. Desta forma o sistema pode recomendar filmes nos quais indivíduos com gostos semelhantes se interessariam, mas não assistiram ainda.

A tabela 4.1 mostra na prática como a filtragem colaborativa pode funcionar. Por exemplo, se quisermos recomendar um produto ao usuário Mauro, procuraremos outros usuários com hábitos de consumo semelhantes. No caso, Paulo e João já compraram produtos que Mauro também comprou (Prod2). Em seguida, recomendamos a Mauro produtos que estes dois outros usuários possuem mas que Mauro ainda não possui, como Prod1 e Prod5.

Tabela 4.1. Recomendação baseada em filtragem colaborativa

Usuário	Prod ₁	Prod ₂	Prod ₃	Prod ₄	Prod ₅	Prod ₆
Paulo		x			x	
João	x	x				
Márcia			x	x	x	
Carlos			x			
Ana	x			x		
Mauro		x				

A filtragem colaborativa apresenta algumas vantagens, como por exemplo a possibilidade apresentar aos usuários recomendações inesperadas. O usuário poderia receber recomendações de itens que não estavam sendo pesquisados de forma ativa. Outra contribuição importante dos sistemas de filtragem colaborativa se refere à possibilidade de formação de comunidades de usuários pela identificação de seus gostos e interesses similares.

Uma questão importante em sistemas colaborativos refere-se à coleta de informações dos usuários, que pode apresentar alguns problemas:

- 1) **Problema do primeiro avaliador:** quando um novo item aparece no banco de dados não existe maneira deste ser recomendado para o usuário até que mais informações sejam obtidas através de outro usuário.
- 2) **Problema de pontuações esparsas:** o objetivo dos sistemas de filtragem colaborativa é ajudar pessoas, focando em documentos lidos ou itens adquiridos. Caso o número de usuários seja pequeno em relação ao volume de informações no sistema existe um grande risco das pontuações tornarem-se muito esparsas.
- 3) **Similaridade:** caso um usuário tenha gostos que variam do normal este terá dificuldades para encontrar outros usuários com gostos similares, sendo assim suas recomendações podem se tornar pobres.

A técnica de Filtragem Colaborativa pode ser separada em três passos:

- a. Calcular o peso de cada usuário em relação à similaridade ao usuário ativo (métrica de similaridade).
- b. Selecionar um subconjunto de usuários com maiores similaridades (vizinhos) para considerar na predição.

- c. Normalizar as avaliações e computar as predições ponderando as avaliações dos vizinhos com seus pesos.

Esta técnica também é chamada de “*k-nearest-neighbor*” ou “*user-based*”. No primeiro passo, a definição da similaridade pode ser realizada através de diversas técnicas, sendo mais comumente usada a correlação de Pearson.

$$w_{a,u} = \frac{\sum_{i=1}^m (r_{a,i} - \bar{r}_a) * (r_{u,i} - \bar{r}_u)}{\sigma_a * \sigma_u} \quad (1)$$

Dado o conjunto de avaliações $r_{a,i}$ do usuário ativo a , e o conjunto de avaliações $r_{u,i}$ do vizinho u , $w_{a,u}$ é o índice de similaridade entre os dois usuários. Note que é preciso mais de uma avaliação em comum para que o índice seja útil, e os resultados variam entre 1 para similaridade total, e -1 para total dissimilaridade.

No passo três, o cálculo da predição é efetuado através da equação (2).

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^n (r_{u,i} - \bar{r}_u) * w_{a,u}}{\sum_{u=1}^n w_{a,u}} \quad (2)$$

O valor da predição $p_{a,i}$ do item i para o usuário ativo a é a média ponderada das avaliações dadas ao item i pelos n vizinhos u do usuário ativo a . A quantidade n de melhores vizinhos - com maiores correlações - é uma escolha de cada sistema que utiliza a filtragem colaborativa.

4.1.3. Filtragem Híbrida

A abordagem da filtragem híbrida procura combinar os pontos fortes da filtragem colaborativa e filtragem baseada em conteúdo visando criar um sistema que possa melhor atender as necessidades do usuário [Herlocker 2000] [Ansari et al. 2000].

Essa abordagem é constituída de vantagens proporcionadas pela filtragem baseada em conteúdo e pela filtragem colaborativa, unindo o melhor das duas técnicas e eliminando as fraquezas de cada uma, conforme expressado pela figura 4.1.

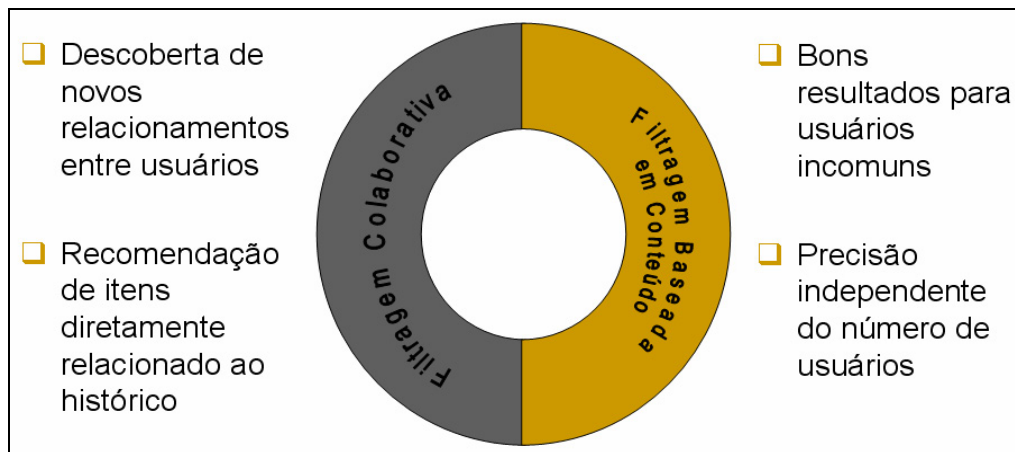


Figura 4.1. Filtragem híbrida

4.2. Descoberta de Conhecimento em Base de Dados (DCBD)

Uma forma de realizar a extração de conhecimento dos bancos de dados dos usuários constitui-se na aplicação de técnicas e ferramentas de mineração de dados. Uma das definições de DCBD encontrada na bibliografia é a que segue:

“...um processo de extração não trivial de informações potencialmente úteis, as quais não são previamente conhecidas e encontram-se implícitas em grandes coleções de dados” [Zaiane 2000].

Quando se trabalha com sistemas de recomendação na Web, a mineração de dados torna-se um recurso importante para a descoberta de relações entre itens, entre usuários e entre itens e usuários. Através da mineração dos arquivos de *log* [Yang et al. 2002], por exemplo, pode-se obter conhecimentos aprofundados a respeito dos usuários que se conectaram a um *website* [Spiliopoulou 2000]. Este conhecimento pode ser utilizado para a personalização da oferta de produtos [Rucker e Polanco 1997], na estruturação de *sites* de acordo com o perfil de cada internauta e personalização [Nulvenna et al. 2000] do conteúdo das páginas.

As próximas seções apresentam diferentes tipos de descoberta de conhecimento aplicáveis nos sistemas de recomendação.

4.2.1. Tipos de descoberta

O desenvolvimento de sistemas de DCBD está relacionado com diversos domínios de aplicações, como por exemplo: análises corporativas, medicina, biologia, mercado, etc [Apte et al. 2002] [Smyth 2002] [Mccarthy 2000]. De acordo com a aplicação algumas técnicas de mineração podem ser mais eficientes do que outras.

a) Regra de associação [Cabena 1997] [Plastino 2001] [Agrawal 1993]

Neste tipo de função cada tupla da base de dados consiste em um conjunto de atributos denominados itens. Cada tupla corresponde a uma transação, e um item pode assumir um valor verdadeiro ou falso (binário). Uma regra de associação constitui-se em um

relacionamento $X \Rightarrow Y$, onde X e Y são conjuntos de itens e a intersecção entre os mesmos constitui-se em um conjunto vazio. Cada regra de associação é associada a um fator de suporte (FSup), e a um fator de confiança (FConf).

O FSup constitui-se na razão do número de tuplas que satisfazem X e Y , sobre o número total de tuplas ($\text{FSup} = |X \cup Y| / N$).

O FConf constitui-se na razão do número de tuplas que satisfazem X e Y sobre o número de tuplas que satisfazem X ($\text{FConf} = |X \cup Y| / |X|$).

A função de extração de regras a partir de um banco de dados consiste em encontrar todas as regras possíveis com FSup e FConf maiores ou iguais a um FSup e FConf especificada pelo usuário. A tabela 4.2 apresenta um exemplo de conjunto de dados que pode ser minerado para identificar se determinado item tem ou não probabilidade de ser adquirido. A primeira coluna apresenta um identificador (ID) da transação, e as outras colunas indicam se um determinado item foi ou não adquirido na transação correspondente. Suponha que o usuário especificou os parâmetros $\text{FSup} = 0.3$ e $\text{FConf} = 0.8$.

Tabela 4.2. Entrada de dados para a descoberta de regras de associação

ID	Artigo1	Artigo2	Artigo3	Artigo4	Artigo5	Artigo6	Artigo7
1	Não	Sim	Não	Sim	Sim	Não	Não
2	Sim	Não	Sim	Sim	Sim	Não	Não
3	Não	Sim	Não	Sim	Sim	Não	Não
4	Sim	Sim	Não	Sim	Sim	Não	Não
5	Não	Não	Sim	Não	Não	Não	Não
6	Não	Não	Não	Não	Sim	Não	Não
7	Não	Não	Não	Sim	Não	Não	Não
8	Não	Não	Não	Não	Não	Não	Sim
9	Não	Não	Não	Não	Não	Sim	Sim
10	Não	Não	Não	Não	Não	Sim	Não

As regras de associação descobertas a partir dos dados da tabela 4.2 são apresentadas a seguir, aplicando-se os valores de FSup e FConf maiores ou iguais aos especificados pelo usuário.

Conjunto de itens freqüentes: Artigo2, Artigo4. $\text{FSup} = 0.3$
 Regra: Se (Artigo2) então (Artigo4). $\text{FConf} = 1$.

Conjunto de itens freqüentes: Artigo2, Artigo5. FSup = 0.3

Regra: Se (Artigo2) então (Artigo5). FConf = 1.

Conjunto de itens freqüentes: Artigo4, Artigo5. FSup = 0.4

Regra: Se (Artigo4) então (Artigo5). FConf = 0.8.

Regra: Se (Artigo5) então (Artigo4). FConf = 0.8.

Conjunto de itens freqüentes: Artigo2, Artigo4, Artigo5. FSup = 0.3

Regra: Se (Artigo2 e Artigo4) então (Artigo5). FConf = 1.

Regra: Se (Artigo2 e Artigo5) então (Artigo4). FConf = 1.

Regra: Se (Artigo2) então (Artigo4 e Artigo5). FConf = 1.

Estas regras descobertas poderiam servir como indicativo de artigos que devam ser oferecidos de forma casada, uma vez que são acessados ou adquiridos juntamente (segundo o que revelou a amostra da base de transações).

Através das regras de associação torna-se possível identificar associações entre diferentes itens que participaram ou não de um conjunto significativo de transações. Entre algoritmos aplicados neste tipo de função destacam-se o Apriori e o Partition [Plastino 2001].

b) Classificação [Cabena 1997] [Quinlan 93]

Nessa função cada uma das tuplas pertence a uma das classes entre um conjunto pré-definido de classes. A classe de uma tupla é indicada por um valor especificado pelo usuário em um atributo meta, ou atributo objetivo.

As tuplas consistem de atributos preditivos e um atributo objetivo, o último indicando a qual classe a tupla pertence. O princípio maior da classificação consiste em descobrir algum relacionamento entre os atributos preditivos e o atributo objetivo, para revelar um conhecimento que possa ser utilizado para prever classes de uma tupla desconhecida, ou seja, que não possui uma classificação.

Um exemplo possível para explicar o uso de classificação é o que segue: supondo que um sistema de recomendação guarde algumas informações sobre os seus usuários (por exemplo, *status* na Universidade, país e área de interesse), seria interessante utilizar esses dados para prever que tipo de usuário estaria mais interessado em qual área de pesquisa.

O sistema poderia então concentrar os esforços de recomendação para classificar os usuários. Para prever se o usuário estará ou não interessado em determinada área pode ser feito o uso de uma árvore de decisão [Holte e Yan 1996].

Considerando apenas os atributos *status*, país e área de interesse dos usuários como relevantes para a previsão, são apresentados na tabela 4.3 os valores dos atributos selecionados, junto com o valor do atributo objetivo e área de interesse.

Tabela 4.3. Entrada de dados para a tarefa de classificação

Status	País	Área de Interesse
Professor	França	Agentes
Pesquisador	Inglaterra	ML
Aluno	França	ML
Pesquisador	Inglaterra	ML
Professor	França	Agentes
Aluno	Alemanha	WebMining
Aluno	Alemanha	WebMining
Pesquisador	Inglaterra	ML
Professor	França	WebMining
Professor	França	ML
Aluno	França	ML

A árvore de decisão gerada com a aplicação de classificação para o exemplo acima poderia apresentar o formato apresentado na figura 4.2.

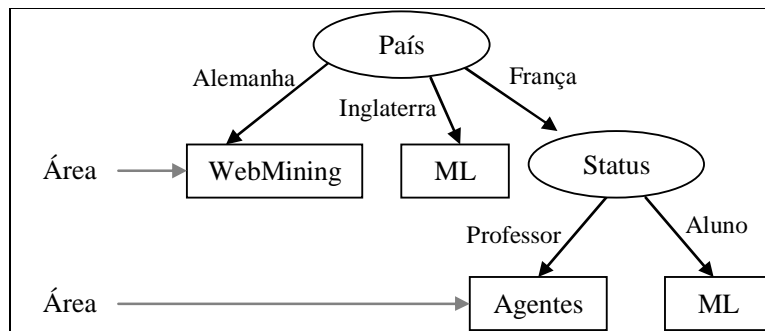


FIGURA 4.2. Árvore de Decisão

O conhecimento descoberto é freqüentemente representado na forma de regras do tipo SE-ENTÃO. Abaixo são descritas as regras obtidas através da classificação:

Se (PAÍS = Alemanha) então Área = WebMining

Se (PAÍS = Inglaterra) então Área = ML

Se (PAÍS = França e Status = Professor) então Área = Agentes

Se (PAÍS = França e Status = Aluno) então Área = ML

Observa-se através destes resultados que fica mais fácil para um sistema de recomendação decidir o que recomendar no caso de um novo usuário cadastrado.

A classificação é um tipo de método classificado como aprendizado supervisionado, uma vez que definimos qual é o atributo objetivo e os atributos preditivos [Cabena 1997]. Em um sistema de recomendação poderíamos utilizar a classificação para identificar a classe de itens novos em categorias pré-definidas ou mesmo para classificar usuários pelos seus perfis.

c) Agrupamento [Cabena 1997]

De uma forma natural as pessoas procuram visualizar os dados segmentados em grupos discretos, onde o que permite a união destes objetos em um mesmo grupo é a similaridade existente entre os mesmos.

Nesta função de DCBD, o algoritmo de agrupamento automaticamente deve criar grupos através de partições de um banco de dados em conjunto de tuplas similares. A partição acontece de forma que as tuplas de valores e atributos semelhantes sejam reunidas nos mesmos grupos. Um agrupamento de qualidade surge onde a similaridade intra-classe é alta e a inter-classe é baixa. O agrupamento é um tipo de aprendizado não-supervisionado uma vez que fica a cargo do algoritmo a definição de quais atributos e seus valores serão utilizados para montar agrupamentos de dados similares.

Um dos métodos de agrupamento mais utilizado é o *k-means*. O primeiro passo no processo é encontrar *k* casos para servirem de “semente” para o processo, por exemplo os *k* primeiros casos de um banco de dados.

O segundo passo no processo é encontrar no banco de dados, para cada um dos registros armazenados, a semente mais próxima dele. A figura a seguir ilustra este processo, onde os pontos escuros representam as sementes, os pequenos círculos transparentes representam os outros registros do banco de dados, e os grandes círculos tracejados representam os grupos.

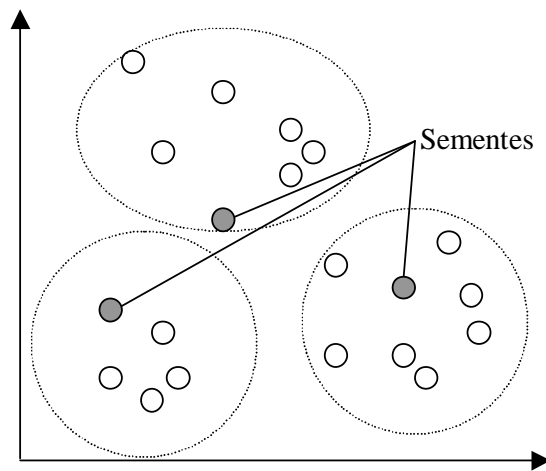


Figura 4.3. Formação inicial de grupos.

Neste momento, todos os registros já foram associados a cada um dos grupos. O próximo passo do algoritmo é determinar os novos centróides a partir de todos os elementos existentes em cada cluster, como exemplifica a figura a seguir.

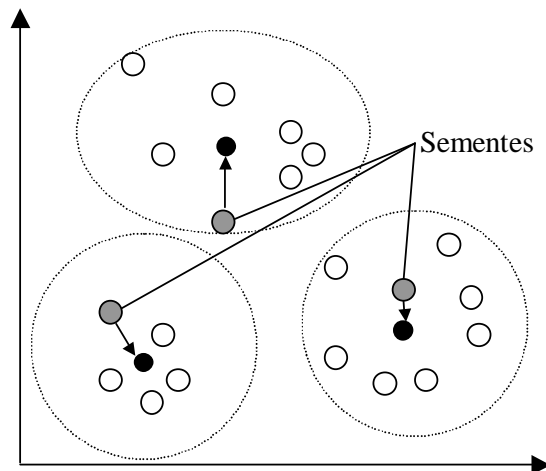


Figura 4.4. Recálculo dos centróides

Este passo é então repetido até que os centróides não precisem ser recalculados. Uma utilidade para agrupamentos poderia ser a reunião de usuários por características comuns ou similares. Um sistema de recomendação pode empregar estes agrupamentos para oferecer itens apropriados de acordo com as características de cada grupo.

4.3. Exemplos de Sistemas de Recomendação

Nesta seção, alguns sistemas de recomendação são descritos, enfatizando suas principais características.

4.3.1. Referral Web [Kautz et al. 1997]

O projeto Referral Web procura identificar e visualizar redes sociais, que de acordo com o autor seriam grupos de pessoas ligadas por atividades profissionais.

Para criar esta rede o sistema utiliza-se de textos já existentes na Web. Uma vez que o usuário entra com seus dados é feita uma busca de documentos que o mencionem. A partir dos documentos recuperados, são extraídos nomes de outros indivíduos citados.

No final é construída uma rede global de citações representada na forma de um grafo. Neste sistema a formação da rede está baseada somente na relação entre usuário, documento e citações, a importância e relevâncias dos trabalhos ou indivíduos não é considerada.

4.3.2. RINGO [Maes e Shardanand 1995]

RINGO é um sistema desenvolvido para recomendação personalizada de música e foi desenvolvido no *Massachusetts Institute of Technology*. O trabalho explora similaridades entre os gostos de diferentes usuários para recomendar itens, baseado no fato de que os gostos das pessoas apresentam tendências gerais e padrões entre gostos, e entre grupos de pessoas. Neste sistema as pessoas descrevem suas preferências musicais através da avaliação de algumas canções. Estas avaliações constituem o perfil dos indivíduos. O sistema usa então estes perfis para gerar recomendações para usuários individuais. Para o seu funcionamento, primeiramente usuários similares são identificados. A partir desta identificação e comparação de perfis o sistema pode prever o quanto o usuário gostaria de um álbum/artista que ainda não foi avaliado pelo mesmo.

Quando o usuário acessa o Ringo pela primeira vez, é apresentado a ele uma lista de 125 artistas. O usuário avalia estes de acordo com o quanto gosta de ouvi-los. Caso o usuário não esteja familiarizado com o artista ou não possua uma forte opinião sobre o mesmo, é solicitado que não avalie o artista para não gerar distorções. Para a pontuação é aplicada uma escala Likert de 7 pontos, sendo o 1 (não gosta), 4 (indiferente) e 7 (adora).

A lista de artistas enviada para os usuários é dividida em duas partes. A primeira parte da lista é gerada a partir dos artistas mais pontuados (*ranking*), o que assegura que um novo usuário tenha a oportunidade de pontuar artistas que outros já tenham pontuado. A outra parte da lista é gerada através de uma seleção randômica a partir de um banco de dados sobre os artistas. Uma vez que o perfil inicial da pessoa tenha sido submetido, o usuário pode solicitar ao Ringo algumas predições:

- 1) sugerir novos artistas/álbuns que o usuário gostaria de obter ou ouvir;
- 2) listar artistas/álbuns que o usuário não gostaria;
- 3) realizar uma predição sobre um artista/álbum específico.

O retorno dado pelo Ringo aos usuários não inclui nenhuma informação em particular sobre a identidade dos outros usuários que contribuíram com as

recomendações. Deve-se enfatizar esta característica em sistemas colaborativos ou de informação social (muitas vezes usados como sinônimo): a identidade de quem avaliou deve ser mantida em segredo.

O sistema Ringo também permite a inserção de comentários (como *feedback* explícito) sobre o produto recomendado. Os próprios usuários podem incluir novos artistas e álbuns no banco de dados.

4.3.3. GroupLens [Riedl et al. 1997]

O GroupLens constituiu-se em um projeto de pesquisa desenvolvido pela Universidade de Minnesota, aplicado à filtragem colaborativa de notícias, auxiliando pessoas a encontrar artigos de interesse entre um grande volume de artigos ofertados.

O Sistema coleta a avaliação dos usuários referente aos artigos lidos (escala de 1-5 pontos) e utiliza esta avaliação para identificar os vizinhos mais próximos com avaliações semelhantes. Desta forma o sistema é capaz de prever se um usuário pode se interessar por um novo artigo baseando-se nas avaliações dos usuários vizinhos.

4.3.4. Fab [Balabanovic e Shoham 1997]

O Fab é um sistema que procura combinar sistemas de filtragem colaborativa e filtragem baseada em conteúdo (abordagem híbrida), procurando eliminar muitas das fraquezas encontradas em cada uma das abordagens. Foi desenvolvido pela Universidade de Stanford em um projeto para recomendação de páginas Web (documentos) para usuários.

Objetivando a construção de uma arquitetura híbrida, perfis de usuários são gerados a partir da análise de conteúdo e comparados para identificar usuários similares para recomendação colaborativa.

O sistema tem uma implementação distribuída e foi construído como parte do projeto da biblioteca digital da Universidade de Stanford. O sistema foi implementado através de dois agentes, um responsável pela coleta dos documentos na Web (*Collection Agent*) e outro responsável pela seleção de páginas (*Selection Agent*) que seriam apresentadas aos usuários como recomendação, conforme a figura 4.5.

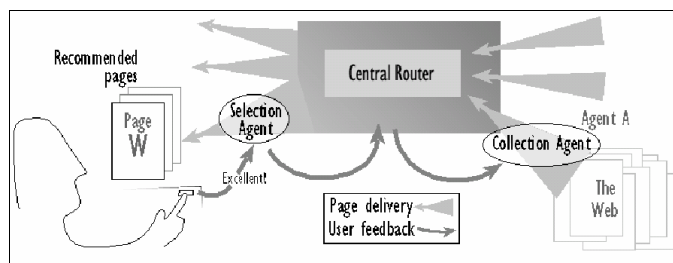


FIGURA 4.5. Visão geral da arquitetura do Fab

Após a recomendação o usuário tem a possibilidade de avaliar o item aplicando uma escala Likert de 1-7 pontos.

4.3.5. Collaborative Recommender Agent CORA [Lueg 1998]

O sistema *Collaborative Recommender Agent CORA* (Christoph apud [Lueg 1998]) é um sistema distribuído assíncrono para filtragem colaborativa ativa de documentos da Web, desenvolvido na Universidade de Zurique. O CORA habilita os usuários a recomendar URLs através de uma simples clique de mouse.

Os usuários eram providos com um agente pessoal de recomendação que monitorava o comportamento no uso do *browser* pelo usuário. O agente reconhecia a página Web que o usuário estava visualizando e apresentava a mesma em uma janela de recomendação especial. Caso o usuário decidisse recomendar a URL para um grupo de usuários pré-definido, a URL poderia ser enviada pela clique de um ícone específico. Assim como o agente monitorava o envio de URLs este também monitorava a chegada de URLs enviadas por outros usuários.

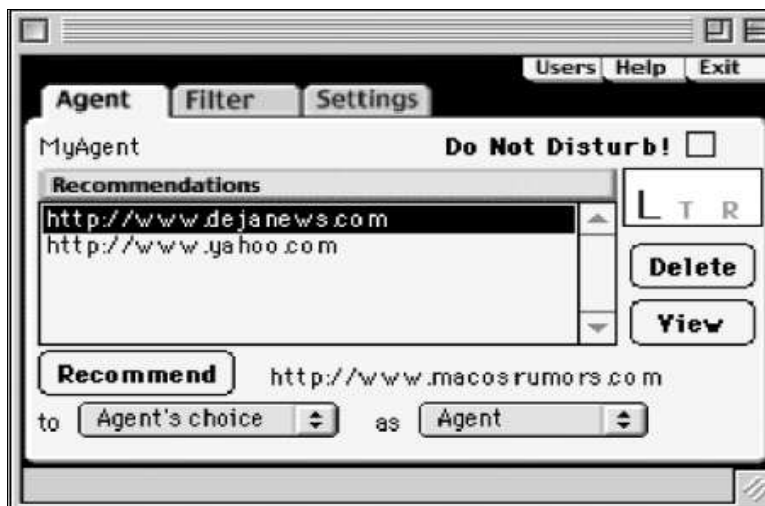


FIGURA 4.6. Interface da ferramenta CORA

Na interface apresentada na Figura 4.6 pode-se observar três letras L,T e R. Cada uma, de acordo com o seu tamanho, procura simbolizar: o montante de *Links* seguidos tendo as páginas recomendadas como ponto de partida; o Tempo utilizado no acesso da página; e um fator especial de Recomendação, o qual é computado com base no uso passado das recomendações. Lueg destaca que sempre que alguém está querendo buscar uma informação de valor, este possui um tempo determinado para aplicar nesta busca, mas este esforço apresenta um limite pequeno [Lueg 1998].

4.3.6. Amazon.comTM

O *site* de comércio eletrônico da Amazon.comTM, disponível em www.amazon.com, possui algumas variações de estratégias de recomendação. Em [Riedl et al. 2000] estas variações são apresentadas, destacando as características de cada uma delas:

- clientes que compraram: são dois tipos de listas de recomendação separadas. A primeira, recomenda livros freqüentemente comprados pelos clientes que adquiriram o livro selecionado; a segunda, recomenda autores cujos livros são

frequentemente comprados por clientes que adquiriram trabalhos de acordo com o autor do livro selecionado;

- suas recomendações: os clientes qualificam os livros lidos em uma escala de 1 a 5, variando de “eu odeio isto” a “eu adoro isto”. Após a avaliação de um conjunto de livros, os clientes podem requisitar recomendações de livros que lhe agradem;
- olhos: permite aos clientes receberem e-mails sobre novos itens que foram adicionados no catálogo. Os clientes selecionam as informações, como autor, título, assunto, entre outros, ou utilizam o formato avançado com operadores booleanos (*and / or*) para filtragem de notificações;
- amazon.com entrega: os clientes selecionam, em caixas de seleção (*checkbox*), os itens de uma lista de categorias e gêneros específica. Periodicamente, os editores da Amazon.comTM enviam as suas últimas recomendações, por e-mail, para os inscritos em cada categoria;
- idéias de presentes da livraria: permite aos clientes receber recomendações dos editores. Os clientes escolhem uma categoria de livros para as quais gostariam de receber algumas sugestões;
- comentários dos clientes: permite aos clientes receberem recomendações, em formato textual, baseadas em opiniões de outros clientes que, por sua vez, também podem ser qualificadas em uma escala de 1 a 5.

4.3.7. eBayTM

Assim como no *site* da Amazon.comTM, o *site* de leilões on-line eBay.comTM, acessível no endereço www.ebay.com, possui mais de uma variação de estratégias de recomendação, também apresentado por [Riedl et al. 2000]. São elas:

- direito de resposta: permite aos compradores e vendedores avaliar o seu parceiro de negócio de acordo com o grau de satisfação da transação de compra. Isto gera uma pontuação que demonstra a confiabilidade de cada negociante. Quanto mais positiva a pontuação, mais confiável é o negociante, e quanto mais negativa a pontuação, menos confiável é o negociante;
- comprador pessoal: permite aos clientes indicar os itens que eles têm interesse em comprar, para que, em uma periodicidade definida, o *site* possa enviar os resultados da busca por esses itens.

5. Arquitetura e aspectos de implementação

Um sistema de recomendação deve funcionar como um servidor capaz de recuperar, para determinado usuário, suas preferências explícitas, implícitas, conteúdo e ofertas relacionados a itens nos quais já mostrou interesse. Tal sistema deve ficar em comunicação com o servidor que apresenta as recomendações ao usuário (e.g. um servidor web).

São dois os principais tipos de arquitetura possíveis em sistemas de recomendação. No primeiro deles, baseado em técnicas de filtragem de informação, não há componente de mineração de dados já que o sistema opera através da análise direta

dos dados dos usuários, sem a construção de um modelo de recomendações generalizado. Diferentes estratégias de recomendação são armazenadas e permitem que sejam feitas diferentes tipos de sugestões (recomendações de presentes, de itens mais populares, etc). Um servidor web faz requisições ao sistema de recomendação, que retorna recomendações conteúdo, serviços e ofertas personalizados.

A figura abaixo mostra a comunicação entre os principais módulos deste tipo de configuração.

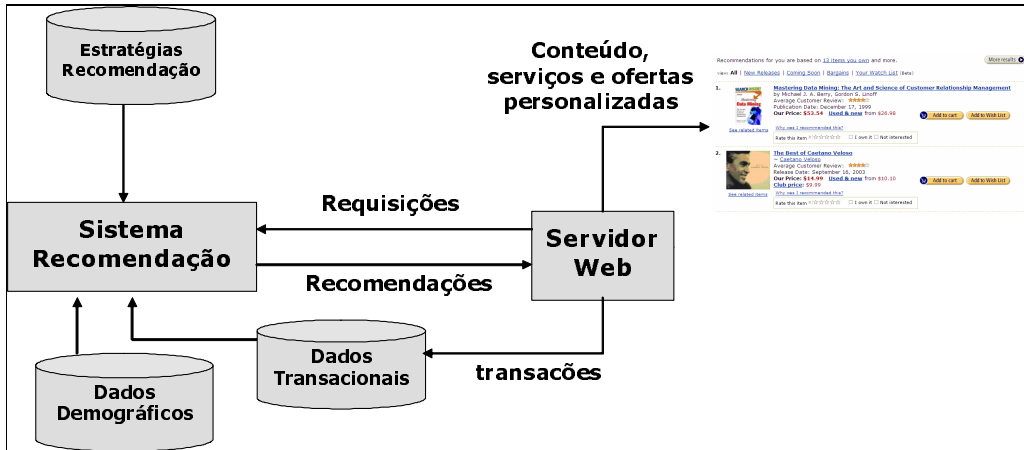


Figura 5.1. Estrutura de Sistema de Recomendação baseado em técnicas de filtragem de informação

No segundo tipo de arquitetura, há um módulo de mineração de dados responsável por encontrar padrões nos dados demográficos e transacionais do usuário, como ilustra a figura abaixo.

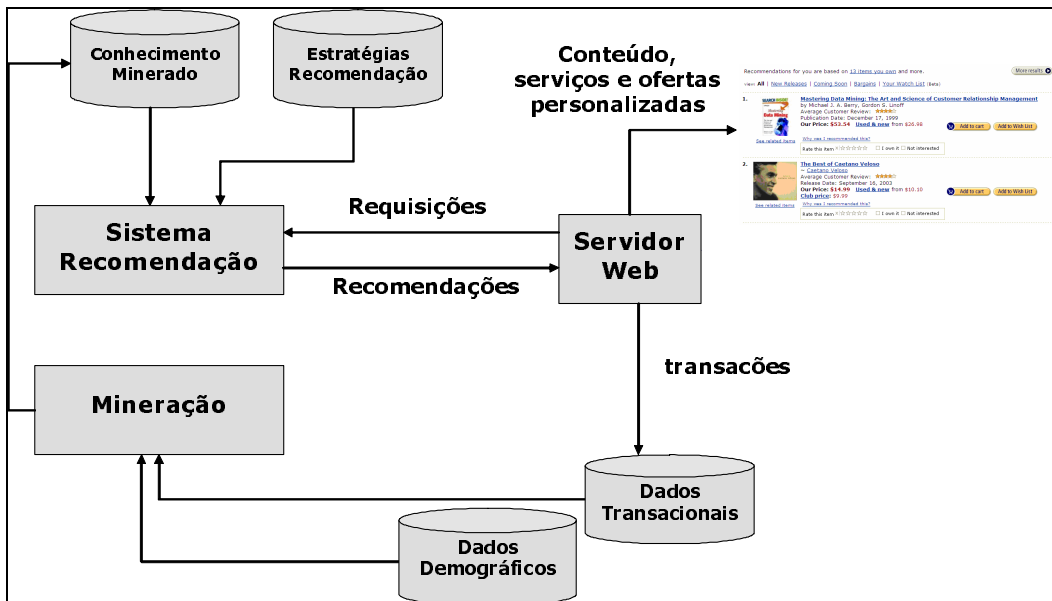


Figura 5.2. Estrutura de Sistema de Recomendação empregando mineração de dados

Os padrões identificados pelo sistema de mineração são armazenados em uma base de conhecimento. Diferentes estratégias de recomendação também são guardadas nesta modalidade de arquitetura e regem o tipo de sugestão que se pode fazer ao usuário. Um servidor web faz requisições ao sistema de recomendação, que retorna recomendações sem interferência direta do sistema de mineração.

As próximas subseções apresentam a arquitetura de um sistema de recomendação fictício baseado em filtragem colaborativa chamado *Recommender*, capaz de identificar clientes, analisar informações e elaborar ofertas precisas de produtos e serviços em diferentes pontos de contato.

5.1. O Sistema *Recommender*

Esta seção apresenta a modelagem de um sistema de recomendação chamado aqui de *Recommender*, cujo objetivo é permitir que clientes e visitantes do website de uma empresa possam receber, no momento exato, recomendações de produtos e serviços adequados aos seus interesses. Estas recomendações são obtidas através de indicações explícitas de preferências ou através da técnica de filtragem colaborativa.

O sistema *Recommender* utiliza uma série de estruturas para:

- armazenar informações sobre preferências dos usuários;
- armazenar informações sobre itens consultados ou adquiridos pelos usuários;
- registrar dados sobre a interação com o usuário durante a sessão;
- representar estratégias de recomendação.

A figura a seguir apresenta um esquema no qual é possível visualizar todas as estruturas utilizadas pelo *Recommender* bem como seus relacionamentos. As próximas seções apresentam cada uma delas.

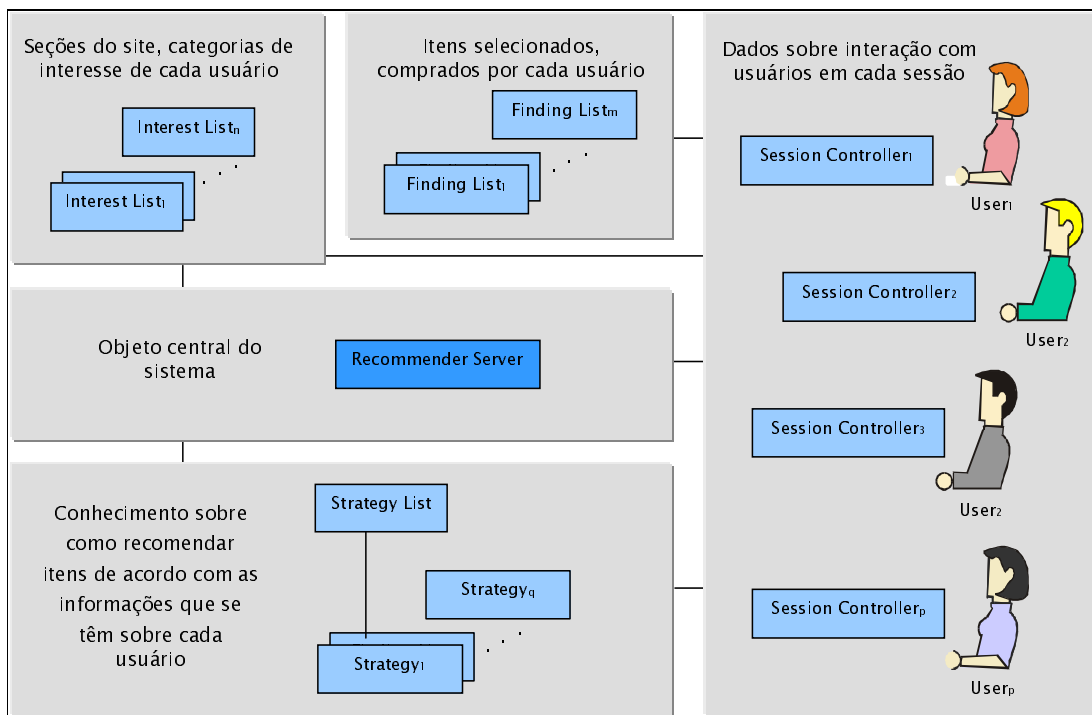


Figura 5.3. Estruturas utilizadas pelo sistema Recommender

5.1.1. Recommender Server

O principal objeto do sistema Recommender é chamado de *RecommenderServer*, sendo utilizado principalmente para dar acesso aos *SessionControllers* (controladores de sessão) e às rotinas de busca de recomendações.

5.1.2. Session Controller

Quando um usuário se conecta a um website, uma sessão é inicializada para ele, durante a qual várias informações são armazenadas sobre sua interação. Por exemplo os itens visualizados, as páginas navegadas, etc. Estas informações são armazenadas no objeto de nome *SessionController*, ou controlador de sessão. No entanto, além de registrar dados sobre a interação com o usuário numa nova sessão, o *SessionController* é também carregado com todas as informações que se coletou no passado sobre um usuário. Por exemplo os itens que foram adquiridos por ele, áreas de preferência no site, entre outros.

5.1.3. Strategy List

Uma estratégia de recomendação contém um conjunto de modelos que definem a forma com que o sistema fará a recomendação de conteúdo, produtos, serviços ou outros. Podemos ter estratégias baseadas em filtragem colaborativa, baseadas em listas armazenadas pelo sistema (mais vendidos, lista de presentes, etc), ou ainda baseadas em preferências explícitas ou implícitas dos usuários.

Portanto, a lista de estratégias guarda vários modelos de recomendação, cada um orientado a um tipo de recomendação específica. Diferentes estratégias podem ser criadas e utilizadas em paralelo em um mesmo website, cada uma estabelecendo como serão feitas as recomendações em determinada seção do sistema. Na página inicial de um website de comércio eletrônico, por exemplo, podemos utilizar uma estratégia para recomendação de livros baseada nas preferências explícitas do usuário. Na mesma página, outra estratégia pode ser empregada para recomendar CDs a partir de uma comparação feita nos dados de compra deste usuário com outros usuários (filtragem colaborativa). Na figura abaixo, podemos ver em uma mesma página uma estratégia de recomendação de livros (1) ao lado de uma estratégia de recomendação de novidades (2).

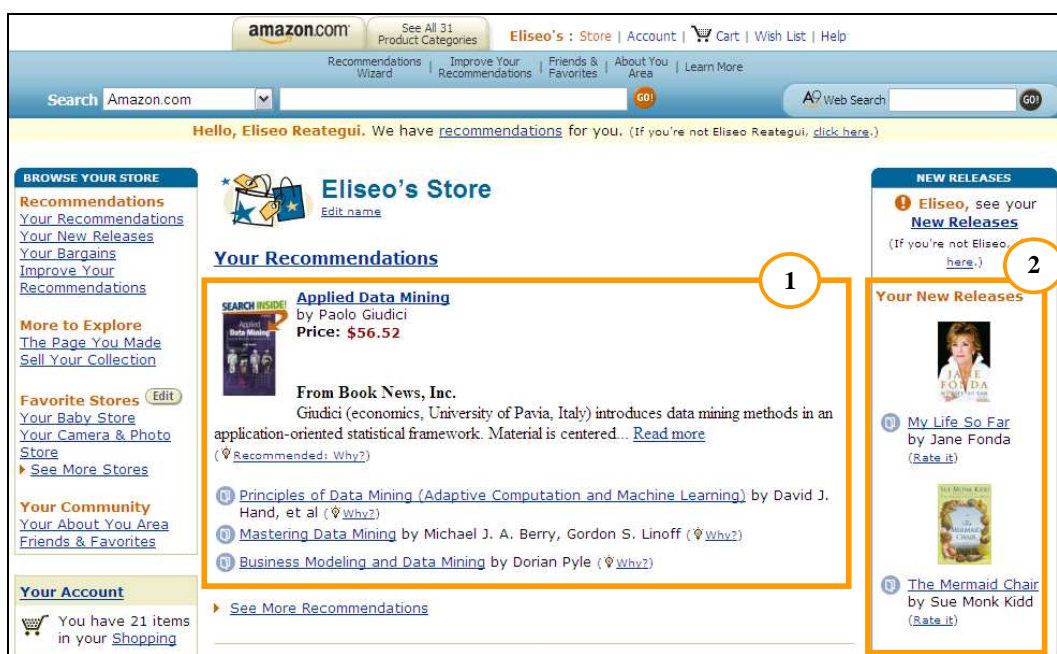


FIGURA 5.4. Estratégias de recomendação

5.1.4. Interest List

Uma *InterestList* (lista de interesses) mantém as preferências de um usuário, que podem ser coletadas implicitamente ou explicitamente. No caso da coleta implícita, na medida em que o usuário navega, seleciona e adquire itens, suas preferências são armazenadas na *InterestList*. Já na coleta explícita, o usuário indica espontaneamente suas preferências e estas são armazenadas em outra *InterestList*. Mais tarde, todas estas preferências podem ser utilizados para se fazer a seleção de conteúdo, de produtos, ofertas ou outros.

5.1.5. Finding List

As *FindingLists* (listas de achados) complementam as informações que se tem sobre os usuários, sendo utilizadas para armazenar características demográficas sobre eles (e.g. faixa etária, sexo, ocupação, ...) ou hábitos de consumo (e.g. itens adquiridos ou colocados num carrinho de compras), ou outro. É através das listas de achados criadas

para um usuário que se pode encontrar recomendações precisas para ele, com base em padrões encontrados em outros usuários similares.

5.2. Personalizando um website

Esta sessão traz um exemplo de como construir uma página personalizada utilizando os objetos e métodos definidos para o sistema Recommender.

Suponha que um usuário qualquer se conecta ao website de uma loja de comércio eletrônico, informando seu usuário e login. O website reconhece o usuário e então monta uma página personalizada para ele, com alguma conteúdo padrão (apresentado para todos os usuários), e duas porções com conteúdo personalizado, como ilustrado na figura 5.4.

Para produzir as recomendação apresentadas na porção (1) da imagem (recomendações a partir dos itens comprados pelo usuário), podemos utilizar os métodos do sistema de recomendação do seguinte modo:

```
// tendo-se o identificador do usuário (userID) e o principal objeto do
// sistema (RecommenderServer), recuperar inicialmente o controlador da
// sessão

Set recommenderSessionController =
recommenderServer.getRecommenderSessionController(userID)

// Cria uma lista vazia para a lista de interesses

interestLists = Array()

// recuperar todas as listas de achados do usuário (e.g. itens comprados,
// itens visitados, etc.)

findingLists = recommenderSessionController.getFindingListsNames()

// fazer a chamada da busca de recomendação, passando o identificador da
// estratégia, das listas de achado e de interesse, bem como do número de
// recomendações desejadas

recommendations = recommenderServer.getRecommendation
("filtragem_colaborativa",findingLists, interestLists, 4)
```

Já as recomendações da porção (2) da imagem (recomendações de novidades), podem ser obtidas do seguinte modo:

```
// tendo-se o identificador do usuário (userID) e o principal objeto do
// sistema (RecommenderServer), recuperar inicialmente o controlador da
// sessão

Set recommenderSessionController =
    recommenderServer.getRecommenderSessionController(userID)

// monta a lista de interesses que vai utilizar

interestLists = Array()
interestLists[1] = "FavoriteBooks"

// Cria uma lista vazia para a lista de achados

findingLists = Array()

// fazer a chamada da busca de recomendação, passando o identificador da
// estratégia, das listas de achado e de interesse, bem como do número de
// recomendações desejadas

recommendations = recommenderServer.getRecommendation
    ("rec_novidades", findingLists, interestLists, 2)
```

6. Privacidade

Para que possamos recomendar itens apropriadamente para usuários de um sistema, devemos constantemente coletar dados sobre eles. Dados comportamentais (navegação, compra,...) e possivelmente demográficos. Esta coleta, como vimos anteriormente, é feita muitas vezes de forma implícita, i.e. sem que o usuário necessariamente perceba que informações sobre ele estão sendo armazenadas na medida em que utiliza o sistema. Esta prática traz algumas questões relacionadas à privacidade.

Pesquisas mostram que a maior parte dos usuários está disposta a fornecer informações suas para que possam receber ofertas personalizadas. No entanto, o censo americano mostra que 75% dos usuários daquele país se preocupam com a possível divulgação de dados que fornecem às empresas [Torres 2004]. Os usuários buscam sempre conhecer os objetivos da coleta de dados e se estes serão fornecidos a terceiros. A figura abaixo mostra uma coleta de dados incentivando a participação do usuário para que as recomendações feitas sejam mais eficientes.

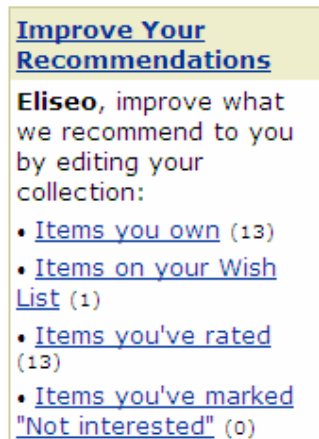


FIGURA 6.1. Coleta de dados para melhoria das recomendações

Portanto, a utilização de sistemas de recomendação não teria problema algum se as empresas que os empregam tivessem uma coleta de dados associada a uma política de privacidade adequada. No entanto, não é incomum encontrar empresas que, além de coletar dados para personalizar o relacionamento com seus clientes, vendam estes dados para outras empresas, uma prática que alimenta o aumento de *spam*. Atitudes como esta, ao invés de estreitar o relacionamento com os clientes, apenas aumentam a distância entre estes e a empresa.

Como atualmente não há leis de privacidade on-line muito claras, é importante que as políticas de privacidade das empresas sejam precisas. A figura abaixo mostra um trecho da página contendo a política de privacidade da loja Amazon.comTM.

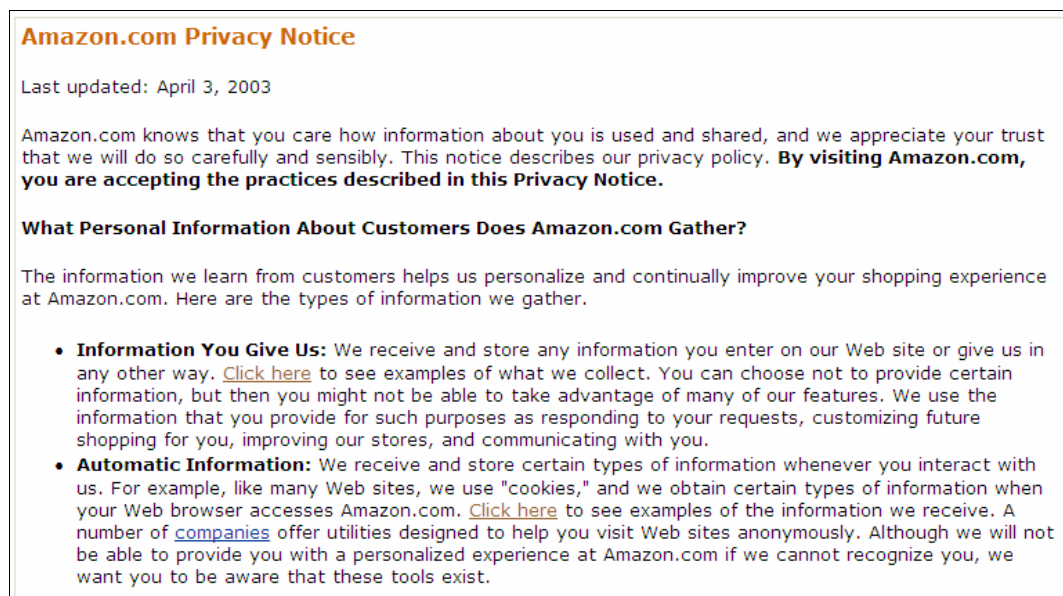


FIGURA 6.2. Política de privacidade da loja Amazon.comTM

Uma política de privacidade não é apenas uma exigência legal, mas também a única forma de garantir que uma empresa agirá com honestidade e protegerá as

informações dos seus clientes. No momento em que existir um padrão e uma forma simples de identificação dos clientes, as políticas de privacidade serão mais eficientes.

Um problema que tem sido levantado com frequência é no caso de dados de clientes que são repassados no momento de compras ou fusões entre empresas. Esta prática é vista como incorreta, e algumas empresas fortes no mundo online (como Amazon e CDNow) têm protegido os dados de seus clientes nestas situações. Outro problema similar ocorreu em 2003 com a companhia americana JetBlue, que forneceu dados de seus clientes ao Departamento de Defesa dos Estados Unidos para que estes pudessem identificar passageiros de alto risco [Torres 2004].

Uma nova alternativa está sendo desenvolvida pela World Wide Web Consortium (W3C) e chamada de Platform for Privacy Preferences (PP3). A PP3 é um protocolo que tem como objetivo principal implementar métodos para definir políticas de privacidade padronizadas, os quais podem ser compreendidas e processadas por computadores. Quando um cliente visita um website a P3P é executada e a mesma pergunta ao cliente quais as informações ele deseja compartilhar com o site, criando assim a política de privacidade. Por exemplo, um cliente talvez deseje informar o seu email e número do telefone com a promessa de que o site somente utilizará essas informações em suas iterações e com a devida permissão.

Algumas empresas estão propondo selos que garantem que a política de privacidade de um website é adequada e cumprida com rigor. A figura 6.3 apresenta um destes selos, fornecidos pela empresa TRUSTe.

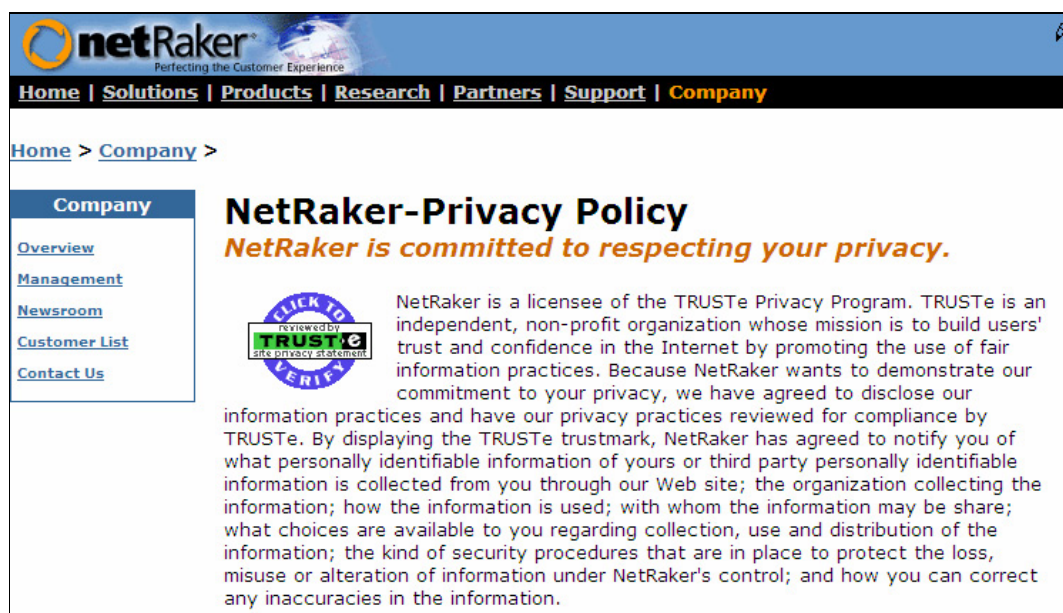


FIGURA 6.3. Selo da empresa TRUSTe no website da empresa NetRaker

Outra solução ao problema é trazida pelo pesquisador Seth Godin [Godin 1999], autor do livro "Marketing de Permissão". A proposta do autor é que jamais se envie mensagens para uma pessoa sem que esta tenha dado permissão explícita para o envio. Além disso, mesmo que a pessoa dê permissão para o envio de material, ela deve poder

interromper este envio com facilidade, a qualquer momento. Através desta abordagem busca-se construir um relacionamento duradouro com os clientes, ao invés de empregar os sistemas de recomendação apenas para alcançar lucratividade rápida.

7. Conclusões e comentários finais

Vimos neste capítulo como os sistemas de recomendação podem ser utilizados para que se possa conhecer melhor os hábitos de consumo e interesses dos clientes, e como este tipo de conhecimento pode ser empregado para personalizar o relacionamento entre a empresa e seus consumidores.

A maior parte dos exemplos apresentados neste curso se referiam à personalização de websites. No entanto, outras aplicações dos sistemas de recomendação são possíveis e encontram cada vez mais mercado. Uma destas aplicações é a de *call-centers*. A maior parte das empresas que lida diretamente com os clientes possui hoje em dia um serviço de *call-center*. Além da infra-estrutura pesada, grande parte do custo de um *call-center* está associado ao pessoal que nele trabalha. Portanto, otimizar este serviço e rentabilizá-lo é algo muito importante. Os sistemas de recomendação permitem, por exemplo, que se possa recomendar a um usuário que está comprando por telefone, produtos que ele provavelmente estará interessado de acordo com as mercadorias já escolhidas.

De posse do perfil do cliente, o atendente tem maior facilidade de estabelecer a comunicação com ele e de efetuar, por exemplo, alguma venda de produto em oferta. Em [Peppers et al. 1999] são apontadas outras vantagens na utilização de serviços de atendimento personalizados:

- Redução do tempo de atendimento em 7%
- Aumento na satisfação do cliente em 8%
- Redução no abandono de comunicações em 19%

A personificação associada à personalização também se mostra uma nova tendência. O conceito de personificação diz respeito ao uso da figura humana na interface do sistema, normalmente na forma de um personagem cujo objetivo é acompanhar o usuário na sua navegação e lhe apresentar recomendações. A utilização da figura humana para melhorar a comunicação com o usuário é baseada na premissa de que a interação com um assistente personificado torna a interface do sistema mais amigável e centraliza a atenção do usuário em torno deste personagem. Várias empresas têm oferecido este tipo de recurso na estruturação de serviços de atendimento online automatizado.

As principais funções às quais um assistente virtual se propõe:

- Responder perguntas de forma sistemática;
- Auxiliar a navegação do usuário, reconhecendo-o, enfatizando sessões e novidades de interesse;
- Coletar dados, solicitando informações de forma gradual;
- Capturar a atenção do usuário, tornando a comunicação com ele mais eficiente.

Além disso, os assistentes virtuais podem também servir como interface para encaminhar recomendações ao usuário, como ilustra a figura abaixo.



Figura 7.1. Recomendações encaminhadas por um assistente virtual

Em [Reategui and Lorenzatti 2005] é apresentada uma aplicação de sistemas de recomendação para ambientes educativos em que um assistente tem a função de responder perguntas dos alunos e recomendar conteúdos a estes, como mostra a figura a seguir.



Figura 7.2. Assistente virtual em ambiente de aprendizagem

A utilização da figura humana em interfaces é um tópico que ainda apresenta controvérsias [Kaasinen 1998]. No entanto, alguns resultados científicos começam a ser apresentados indicando que a personificação associada à personalização aumenta o envolvimento do usuário e facilita sua interação com o sistema [Blom e Monk 2001]. Resultados comerciais igualmente promissores também já são apontados. Em um artigo do New York Times, publicado em setembro do ano passado [Wallace 2000], é relatado o caso de uma assistente virtual que auxilia o usuário a comprar arranjos de uma loja de flores pela internet, recomendando-lhe diversos tipos de arranjo de acordo com certas preferências apontadas. A loja de flores registrou um aumento nas vendas de até três

vezes quando usuários consultam a assistente de vendas. Uma das razões apontadas para o sucesso do site é que pessoas querem comprar de pessoas, e não de máquinas.

A necessidade de personalizar o relacionamento com os clientes é cada vez mais crescente nos dias de hoje. Ela é na verdade uma forma de volta ao passado, onde o vendedor de um pequeno mercado conhecia cada um de seus clientes e conseguia, de forma bastante eficiente, lembrar-se das preferências de cada um para fazer recomendações que aumentavam a lucratividade do negócio e promoviam a fidelização.

Hoje em dia, para o vendedor de uma grande corporação, não é possível lembrar dos principais interesses dos clientes bem como dos itens por eles adquiridos nas últimas compras. Por isso, os bancos de dados, as ferramentas de mineração e recomendação servem como memória da empresa, provendo informações valiosas relacionadas aos hábitos de consumo e preferências dos consumidores. A utilização destas informações na comercialização de produtos não implica na substituição do homem por computadores com maior capacidade de armazenamento de informações, mas sim na adaptação de um tipo de serviço que era prestado por comerciantes e que, hoje em dia, não é mais possível dado o crescente tamanho da população e, conseqüentemente, o número de consumidores.

References

- Agrawal, R. et al. (1993) "Mining Association Rules between sets of items in large databases", In: International Conference Management of Data (SIGMOD-93). Proceedings..., p.207-216.
- Ansari, A. et al. (2000). Internet Recommendation Systems. Journal of Marketing Research, v.37, n.3, p. 363-375, Aug.
- Apte, C. et al. (2002). Business Application of Data Mining. Communications of the ACM, New York, v.45, n.8, p. 49-53, Aug.
- Balabanovic, M.; Shoham, Y. (1997). Fab: Content-Based, Collaborative Recommendation. Communications of the ACM, New York, v.40, n.3, p. 66-72, Mar.
- Belvin, N. J.; Croft, W. B.(1992). Information Filtering and Information Retrieval: two sides of the same coin?. Communications of the ACM, New York, v.35, n.12, p. 29, Dec.
- Blom, J. e Monk, A. (2001). "One-to-one e-commerce. who's the one?". Proceedings of CHI 2001: Conference on Human Factors in Computing Systems. Seattle, Washington, 31 March - 5 April.

- Cabena, P. et al. (1997). *Discovering data mining: from concept to implementation*, Prentice-Hall.
- Foltz, P. W. ; Dumais, S. T. (1992). *Personalized Information Delivery: An Analysis of Information Filtering Methods*. *Communications of the ACM*, New York, v.35, n.12, p. 51-60, Dec.
- Godin, S. (1999) *Permission Marketing: Turning Strangers Into Friends And Friends Into Customers*. Simon & Schuster.
- Goldberg, D. et al. (1992). *Using collaborative filtering to weave an information Tapestry*. *Communications of the ACM*, New York, v.35, n.12, p. 61-70, Dec.
- Herlocker, J. L. (2000) "Understanding and Improving Automated Collaborative Filtering Systems", Tese de Doutorado (Doutorado em Ciência da Computação), University of Minnesota, Minnesota.
- Holte, R. C. ; Yan, J. Y. (1996). "Inferring What a User is Not Interested", In: *Advances in Artificial Intelligence, Proceedings...*, p. 159-171.
- Kaasinen, E. (1998). "Usability Issues in Agent Applications: What Should the Designer be Aware of". *USINACTS 01.06.98*, VTT Information Technology, Finland.
- Kautz, H. et al. (1997). *Referral Web: combining social networks and Collaborative filtering*. *Communications of the ACM*, New York, v.40, n.3, p. 63-65, Mar.
- Loeb, S.; Terry, D. (1992). *Information Filtering*. *Comumunications of ACM*, New York, v.35, n.12, p.26, Dec.
- Lueg, C. (1998) "Considering Collaborative Filtering as Groupware: Experiences and Lessons Learned", In: *2nd. International Conference on Practical Aspects of Knowledge Management, Proceedings...*, Basel, Switzerland, p. 29-30.
- Maes, P.; Shardanand, U. (1995) "Social information filtering: Algorithms for automating "word of mouth", In: *Human Factors in Computing Systems. Proceedings...*, 1995, p. 210-217.
- Manber, Udi et al. (2000). *Experience with Personalization on Yahoo! Communication of the ACM*, New York.

- Mccarthy, J. (2000). Phenomenal Data Mining. Communications of the ACM, v.43, n.8, p. 75-79, Aug.
- Montaner, M et al. (2003). A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review. Netherlands : Kluwer Academic Publishers, pp. 285-330, Aug.
- Nulvenna, M. et al. (2000). Personalization on the Net Using Web Mining. Communications of the ACM, v.43, n.8, pp. 122-125, Aug.
- Peppers, D. et al. (1999). The One to One Fieldbook : The Complete Toolkit for Implementing a 1 To 1 Marketing Program, Bantam Books.
- Plastino, A. (2001) "Regras de Associação e Algoritmos de Mineração de Dados", Mini-Curso 2, SBBDD, 2001. Instituto Militar de Engenharia. Rio de Janeiro: Rio de Janeiro.
- Quinlan, J. R. (1993). C4.5: programs for machine learning, San Mateo: Morgan Kaufmann.
- Reategui, E. e Lorenzatti, A. (2005). "Um Assistente Virtual para Resolução de Dúvidas e Recomendação de Conteúdo". Proceedings V Encontro Nacional de Inteligência Artificial (ENIA 2005). São Leopoldo, RS, Julho.
- Resnick, P. e Varian, H. R. (1997). Recommender Systems. Communications of the ACM, New York, v.40, n.3, pp. 55-58, Mar.
- Riedl, J. et al. (1997). Grouplens: Applying Collaborative Filtering to Usenet News. Communications of the ACM, New York, v.40, n.3, pp. 77-87, Mar.
- Riedl, J. et al. (1999) "Combining Collaborative Filtering with Personal Agents for Better Recommendations", In: Proceedings of AAAI, Proceedings..., vol. 35, pp.439-446. Press.
- Riedl, J. et al. (2000). Electronic Commerce Recommender Applications. Journal of Data Mining and Knowledge Discovery, vol. 5 nos. 1/2, pp. 115-152, Aug.
- Rucker, J. ; Polanco, M. J. (1997). Siteseer: Personalized Navigation for the Web. Communications of the ACM, New York, v.40, n.3, pp. 73-75, Mar.

Smyth, P. (2002). Business Applications of Data Mining. Communications of the ACM, v.45, n.45, p. 49-53, Aug.

Spiliopoulou, M. (2000). Web Usage Mining for WebSite Evaluation. Communications of the ACM, New York, v.43, n.8, pp. 127-134, Aug.

Torres, R. Personalização na Internet. Editora Novatec, 2004.

Wallace, D. J. (2000). "She's Only Code and Pixels, but She Can Help You Shop". The New York Times, September 20.

Yang, H. et al. (2002) "On the use of constrained association for Web log mining", In: Web Mining for Usage Patterns & User Profiles, WebKDD, 4., 2002, Edmonton. Proceedings... Alberta, Canadá: ACM SIGKDD, pp. 100-114.

Zaiane, O. R. (2000) "Web Mining: Concepts, Practices and Research", In: Simpósio Brasileiro de Banco de Dados, SBBD, 15., João Pessoa. Tutorial... João Pessoa: CEFET-PB; Porto Alegre: PUCRS, v.43, n.8., pp. 410-474.

Apêndice I - Métodos da API do Sistema Recommender

A interface do sistema Recommender utiliza apenas dois objetos, dando acesso a todas as funcionalidades do sistema. São eles:

- *RecommenderServer*
- *RecommenderSessionController*

Esta seção descreve os métodos da API para cada um destes objetos.

RecommenderServer

Init

public void init (Configurator configurator)

configurator: arquivo “.rpr” contendo o caminho do arquivo de configuração do sistema

Método que inicializa o servidor de recomendação e define o caminho do arquivo de configuracao do sistema, arquivo criado através do GP Explorer.

getRecommenderSessionController

public RecommenderSessionController

getRecommenderSessionController (int userId)

userId: identificador do usuário

Método que retorna um objeto controlador de sessão e o associa ao usuário com identificador *userId*. Se o usuário já possuir um controlador de sessão ativo, nenhum objeto é criado e o objeto ativo é retornado.

RecommenderSessionController

getRecommendation

public int[] getRecommendation(int strategyId, String[]

findingListName, String[] interestListName, int size)

strategyId: Identificador da estratégia a ser utilizada na busca de recomendações

findingListName: Array com identificadores das listas de achados a serem utilizadas

interestListName: Array com identificadores das listas de interesses a serem utilizadas

size: Número de recomendações a serem retornadas

Esta função retorna um array com os identificadores dos itens a serem recomendados a um usuário. As informações sobre o usuário que devem ser utilizadas na seleção de itens para recomendação são definidas pelo parâmetro *findingListName*, que determina as listas de achados a serem utilizadas

(compras, seções visitadas, dados demográficos, etc). O parâmetro `interestListName` permite que se restrinja as recomendações a itens que se encontram dentro de certas categorias (as que se encontrarem dentro das listas de interesses especificadas, e.g. CDs, livros, etc). O número de recomendações retornado é definido pelo parâmetro `size`. Caso o sistema não encontre nenhum item para ser recomendado, então a lista de itens default do sistema é utilizada, lista definida na fase de criação do projeto do Recommender, utilizando a ferramenta GP Explorer.

clearFindings

public void clearFindings (String findingListName)

findingListName: Nome da lista de achados a ser zerada

Remove da lista com nome de `findingListName` todos os achados.

clearFindings

public void clearFindings (String findingListName)

findingListName: Nome da lista de achados a ser zerada

Remove todos os achados da lista com nome de *findingListName*.

public void clearFindings ()

Remove todos os achados de todas as listas de achados.

clearInterests

public void clearInterests (String interestListName)

interestListName: Nome da lista de interesses a ser zerada

Remove todos os interesses da lista com nome de *interestListName*.

public void clearInterests ()

Remove todos os interesses de todas as listas de interesses.

clearTemporaryConstraints

public void clearTemporaryConstraints()

Remove todas as restrições da lista de restrições temporárias, lista que tem como função impedir que determinados itens sejam recomendados a um usuário.

getFindingListsNames

public String[] getFindingListNames()

Retorna um array com os nomes de todas as listas de achados presentes no sistema.

getFindings

public int[] getFindings(String findingListName)

findingListName: Nome da lista de achados

Função que retorna um array com os identificadores de todos os achados da lista especificada por *findingListName*. Se a lista não existir ou for nula (= null), então o array retornado tem tamanho zero.

public int[] getFindings()

Retorna um array com os identificadores de todos os achados de todas as listas de achados do sistema. Se não houver nenhuma lista de achado, então o array retornado tem tamanho zero.

getInterestListsNames

public String[] getInterestListNames()

Retorna um array com os nomes de todas as listas de interesses presentes no sistema.

getInterests

public int[] getInterests(String InterestListName)

InterestListName: Nome da lista de achados

Função que retorna um array com os identificadores de todos os interesses da lista especificada por *InterestListName*. Se a lista não existir ou for nula (= null), então o array retornado tem tamanho zero.

public int[] getInterests()

Retorna um array com os identificadores de todos os interesses de todas as listas de interesse do sistema. Se não houver nenhuma lista de interesse, então o array retornado tem tamanho zero.

putFinding

public void putFinding(int idFinding, String findingListName)

idFinding: Identificador do achado a ser inserido na lista

findingListName: Nome da lista de achados

Método que adiciona o achado cujo identificador é passado como parâmetro (*idFinding*) à lista de achados com o nome especificado por *findingListName*.

public void putFinding (String attrName, String value, String findingListName)

attrName: nome do atributo que permite identificar achado a ser inserido na lista

value: valor do achado a ser inserido na lista

findingListName: Nome da lista de achados

Método que adiciona o achado, cujo nome e valor são passados como parâmetro, à lista de achados especificada por *findingListName*. Se nenhum achado for encontrado para os parâmetros especificados, nenhuma ação é executada.

putInterest

public void putInterest (int idInterest, String interestListName)

idInterest: Identificador do interesse a ser inserido na lista

InterestListName: Nome da lista de interesses

Método que adiciona o interesse cujo identificador é passado como parâmetro (*idInterest*) à lista de interesses com o nome especificado por *InterestListName*.

putTemporaryConstraint

public void putTemporaryConstraint (int idFinding)

idFinding: Identificador do achado a ser colocado na lista de restrições

Método que adiciona o achado identificado por *idFinding* à lista de restrições temporárias do sistema associada ao controlador de sessão (SessionController). Quando colocado nesta lista, um item não é recomendado ao usuário.

public void putTemporaryConstraint(String attrName, String value)

attrName: nome do atributo que permite identificar achado a ser inserido na lista

value: valor do achado a ser inserido na lista

Método que adiciona o achado identificado pelo nome do atributo e valor (*attrName/value*) à lista de restrições temporárias associada ao controlador de sessão (SessionController). Quando colocado nesta lista, um item não é recomendado ao usuário.

removeFinding

public void removeFinding(int idFinding, String findingListName)

idFinding: Identificador do achado a ser removido da lista

findingListName: Nome da lista da qual o achado deve ser removido

Remove o achado identificado por *idFinding* da lista especificada por *findingListName*. Se o achado não existe ou for igual a *null*, então nenhuma ação é executada.

public void removeFinding (String attrName, String value, String findingListName)

attrName: nome do atributo que permite identificar achado a ser removido da lista

value: valor do achado a ser inserido na lista

findingListName: Nome da lista de achados

Método que remove o achado identificado pelo nome do atributo e valor (*attrName/value*) da lista identificada por *findingListName*. Se o achado não existe ou for igual a *null*, então nenhuma ação é executada.

removeInterest

public void remove (int idInterest, String interestListName)

idInterest: Identificador do interesse a ser inserido na lista

InterestListName: Nome da lista de interesses

Método que remove o interesse cujo identificador é passado como parâmetro (*idInterest*) da lista de interesses com o nome especificado por *InterestListName*.

removeTemporaryConstraint

public void removeTemporaryConstraint(int idFinding)

idFinding: Identificador do achado a ser removido da lista

Remove o achado identificado por *idFinding* da lista de restrições associada ao controlador de sessão (*SessionController*).

public void removeTemporaryConstraint (String attrName, String value)

attrName: nome do atributo que permite identificar achado a ser removido das restrições

value: valor do achado a ser removido da lista

Método que remove o achado identificado pelo nome do atributo e valor (*attrName/value*) da lista de restrições associada ao controlador de sessão (*SessionController*).