ALEXSANDRA DA SILVA LÁZARO

ANÁLISE E SELEÇÃO DE ALGORITMOS DE FILTRAGEM DE INFORMAÇÃO PARA SOLUÇÃO DO PROBLEMA *COLD-START ITEM*

LAVRAS - MG

2010

ALEXSANDRA DA SILVA LÁZARO

ANÁLISE E SELEÇÃO DE ALGORITMOS DE FILTRAGEM DE INFORMAÇÃO PARA SOLUÇÃO DO PROBLEMA *COLD-START ITEM*

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para obtenção do título de Bacharel em Sistemas de Informação.

Orientadora:

M.Sc. Juliana Galvani Greghi

Co-orientadora:

Dra. Jerusa Marchi

LAVRAS – MG

2010

ALEXSANDRA DA SILVA LÁZARO

ANÁLISE E SELEÇÃO DE ALGORITMOS DE FILTRAGEM DE INFORMAÇÃO PARA SOLUÇÃO DO PROBLEMA *COLD-START ITEM*

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para obtenção do título de Bacharel em Sistemas de Informação.

APROVADA em ____ de _____ de _____

Dr. Ahmed Ali Abdalla Esmin UFLA

M.Sc. Cristiano Leite de Castro UFLA

M.Sc. Juliana Galvani Greghi

Orientadora

Co-orientadora:

Dra. Jerusa Marchi

LAVRAS - MG

2010

A Dionísio, meu pai.

A Terezinha, minha mãe.

Aos meus irmãos, Diógenes e Dionísio.

DEDICO

AGRADECIMENTOS

Meu especial agradecimento aos meus pais, Dionísio e Terezinha, e irmãos, Diógenes e Dionísio, pelo carinho, apoio e principalmente o companheirismo.

Agradeço aos professores do Departamento de Ciência da Computação da Universidade Federal de Lavras pelos conhecimentos que me foram transmitidos, e especialmente aos professores Ahmed Ali Abdalla Esmin e Cristiano Leite de Castro pelas sugestões e críticas que ajudaram a engrandecer este trabalho.

Agradeço a professora Dra. Jerusa Marchi, pelas orientações no início do desenvolvimento deste trabalho, pela acolhida calorosa sua e de sua família durante minha estada na cidade de Florianópolis, e também por me incentivar e acompanhar o desenvolvimento deste trabalho mesmo estando à distância.

Agradeço a minha orientadora, professora M.Sc. Juliana Galvani Greghi, que sempre esteve à disposição para me ouvir e responder as minhas dúvidas, e também pelas correções e sugestões que só engrandeceram este trabalho.

Agradeço a equipe da empresa Chaordic Systems primeiramente por ter aceitado a parceria de trabalho e a confiança que foi depositada em mim. Também agradeço a recepção e atenção recebida durante minha visita à empresa, o acompanhamento e conselhos que me ajudaram na realização deste trabalho.

Agradeço aos alunos da primeira turma do curso de Sistemas de Informação da Universidade Federal de Lavras pelos momentos agradáveis de estudo e em especial a Carla, Christiane, Clayton, Juliana Alves, Juliana Villas Boas Magrinelli, Maisa e Mariana pela amizade.

Resumo

Atualmente torna-se difícil para as pessoas gerenciar o crescente volume de informação disponível na Internet. Exemplo disso ocorre quando um consumidor precisa escolher dentre uma grande quantidade de produtos disponíveis em *websites* de comércio eletrônico. Para ofertar seus produtos e ajudar consumidores na escolha do que lhes seja mais relevante, empresas de comércio eletrônico utilizam os sistemas de recomendação. Tais sistemas, em geral, fazem uso de técnicas de filtragem de informação para gerar recomendações relevantes. Porém, quando não existem ou são escassos os dados (avaliações, compras, buscas, etc.) sobre os itens, os algoritmos de filtragem não apresentam bons resultados. Este problema é denominado *Cold-Start Item*. Este trabalho realiza um estudo detalhado dos algoritmos de filtragem de informação mais relevantes da última década que possam solucionar o problema *Cold-Start Item* e apresenta uma análise destes algoritmos visando identificar os mais promissores. Este trabalho vem sendo desenvolvido em parceria com a empresa Chaordic Systems de Florianópolis e os resultados aqui apresentados serão utilizados no desenvolvimento de algoritmos para um sistema de recomendação desenvolvido pela empresa.

Palavras-chave: Sistemas de Recomendação, Filtragem de Informação, Filtragem Baseada em Conteúdo, Filtragem Colaborativa, Filtragem Híbrida e problema *Cold-Start Item*.

Abstract

It is currently difficult for people to manage the growing volume of information available on the Internet. One example occurs when a consumer must choose among a lot of products available in e-commerce website. To offer their products and help consumers to choose the most relevant to them e-commerce companies use recommender systems. Such systems generally make use of techniques of information filtering to generate relevant recommendations. But when there are none or few data (ratings, purchases, searches, etc.) about items, the filtering algorithms do not provide good results. This problem is called *Cold-Start Item*. This work makes a detailed study of the most relevant information filtering algorithms of the last decade that can solve the *Cold-Start Item* problem and presents a analysis of these algorithms in order to identify the most promising one. This work is being developed in partnership with the company Chaordic Systems of Florianópolis and the results presented here will be used to develop algorithms for an recommender system developed by the company.

Keywords: Recommender Systems, Information Filtering, Content-Based Filtering, Collaborative Filtering, Hybrid Filtering and *Cold-Start Item* problem.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

CAR	Cross-level Association Rule
CE	Comércio Eletrônico
CLARE	Cross-Level Association RulEs
FARAMS	Fuzzy Association Rule Mining and Multiple-level Similarity
FBC	Filtragem Baseada em Conteúdo
FC	Filtragem Colaborativa
FCH	Filtragem Colaborativa Híbrida
FH	Filtragem Híbrida
FI	Filtragem de Informação
pLSA	probabilistic Latent Semantic Analysis
RA	Regras de Associação
RNA	Redes Neurais Artificiais
SR	Sistemas de Recomendação
SVD	Singular Value Decomposition
TF-IDF	Term Frequency–Inverse Document Frequency

SUMÁRIO

1 INTRODUÇÃO

O acesso à informação é facilitado pelo uso da Internet, que permite a localização de informação de maneira rápida e a qualquer momento. Exemplo disso são os mecanismos de busca, como Google[1] e Yahoo![2] utilizados por quem precisa encontrar informações que tratam de um interesse específico. Através do correio eletrônico a Internet também aumentou e facilitou a troca de informação entre as pessoas, pois tal meio de comunicação possui como uma de suas características básicas a possibilidade de enviar uma mesma mensagem para vários outros endereços eletrônicos ao mesmo tempo.

Dessa forma, a quantidade de informação armazenada e distribuída na Internet cresce cada dia mais. Os mecanismos de busca retornam aos usuários um volume excessivo de informação, deixando sob sua responsabilidade a análise e a seleção das informações que julgar importantes. A troca de mensagens entre os usuários de correios eletrônicos só tende a aumentar a quantidade de informação enviada e recebida pelas pessoas, aumentando o envio de *spam*[3] ou mensagens que possuem conteúdo impróprio, tais como propagandas ou vírus. Estes exemplos mostram que cada vez mais os usuários despendem tempo com a leitura, seleção e classificação de um grande volume de informação, tornando-se importante a utilização de ferramentas de filtragem automática.

[1] www.google.com
[2] www.yahoo.com
[3] Termo usado para referir-se aos *e-mails* não solicitados, que geralmente são enviados para um grande número de pessoas.

1.1 Motivação

Desde seu surgimento, a Internet é usada para pesquisa, troca de mensagens através do correio eletrônico e também para processos de compra, venda e troca de produtos através do comércio eletrônico (CE), além de servir para muitas outras aplicações.

Ao utilizar a Internet como canal de comercialização não só a empresa, mas também os consumidores são beneficiados. No que tange a empresa, a mesma torna-se acessível de qualquer lugar, aumentando o alcance de consumidores, disponibilizando uma enorme quantidade de produtos e operando todos os dias do ano, fatos que dificilmente poderiam ser alcançados pelo método de venda tradicional. No que diz respeito aos consumidores, os mesmos beneficiam-se com maior comodidade podendo comprar de qualquer lugar (trabalho, residência, etc.) a qualquer hora, ter acesso a muitas informações sobre os produtos como preço, características, opiniões de outros usuários, e navegar pela loja através de diferentes categorias de produtos. Este benefício, entretanto, pode vir a tornar-se um problema para o consumidor, que diante de tantas opções não consegue encontrar algo específico.

Para ajudar os consumidores a encontrar produtos que sejam de sua preferência as empresas de CE utilizam os sistemas de recomendação (SR), que coletam informações dos usuários para descobrir suas preferências e assim ofertar produtos e serviços que lhe sejam relevantes. Os SR utilizam a técnica de Filtragem de Informação (FI) para extrair as relações e similaridades existentes entre produtos, entre consumidores e entre produtos e consumidores, dentre elas: (i) Filtragem Baseada em Conteúdo (FBC) que utiliza o perfil do usuário ou características dos produtos para identificar produtos similares aos que o usuário indicou preferência no passado; (ii) Filtragem Colaborativa (FC) que utilizando

dados como histórico de compras, visualização ou avaliações de produtos, encontra relações existentes entre usuários, como interesses em comum (ou não), e também relações entre os itens, como compras conjuntas e (iii) Filtragem Híbrida (FH) que combina a FBC com a FC para fazer recomendações ao usuário.

Os SR apresentam um problema denominado *Cold-Start Item*, que ocorre devido a escassez ou falta de dados sobre produtos, o que impede os SR de gerarem recomendações relevantes aos usuários. Logo a motivação para realização deste trabalho, que busca encontrar soluções para o problema *Cold-Start Item* surgiu da parceria com a empresa Chaordic System cujos SR desenvolvidos sofrem do problema citado.

Este trabalho contribui para a empresa Chaordic Systems, uma vez que ela poderá aprimorar os algoritmos de FI já desenvolvidos por ela e oferecer SR mais eficientes para seus clientes. Em conseqüência disso as lojas de CE clientes da Chaordic Systems serão beneficiadas, pois conseguir recomendar produtos que sofrem do problema *Cold-Start Item* aos seus clientes fará com que aumente as vendas deste produto.

1.2 Objetivos

Este trabalho tem como objetivo encontrar algoritmos de FI que possam solucionar o problema *Cold-Start Item*, que então serão utilizados pela empresa Chaordic Systems[4].

Os objetivos específicos deste trabalho são: (i) realizar uma estudo detalhado dos algoritmos de FI mais relevantes da última década para solução do problema *Cold-Start Item*; (ii) fazer uma análise de cada um dos algoritmos

[4] www.chaordicsystems.com

identificados e (iii) selecionar os algoritmos mais promissores para solução do problema apresentado.

1.3 Resultados esperados

Espera-se que dentre as soluções mais promissoras para o problema *Cold-Start Item* encontradas e analisadas neste trabalho, uma delas seja escolhida e implementada pela empresa Chaordic Systems. E uma vez implementada espera-se que estas melhorem a qualidade e precisão da recomendação de produtos.

1.4 Organização

O capítulo 2 define os SR e também mostra as principais técnicas de FI que podem ser aplicadas aos SR. No capítulo 3 é apresentado o problema *Cold-Start*, e no capítulo 4 a metodologia aplicada a este trabalho. A descrição e análise dos diferentes algoritmos de FI levantados para solução do problema *Cold-Start Item* são apresentadas no capítulo 5, bem como o resultado deste trabalho. Por fim, no capítulo 6 são apresentadas as principais conclusões acerca deste trabalho.

2 REFERENCIAL TEÓRICO

2.1 Sistemas de Recomendação

O primeiro SR, Tapestry (GOLDBERG et al., 1992), foi desenvolvido no início dos anos 90 como forma de lidar com grande quantidade de *e-mails* e mensagens postadas a grupos de notícias. Atualmente o grande foco de utilização dos SR são lojas de CE. Segundo os autores Schafer, Konstan e Riedl (1999) em 1999 os SR já deixavam de ser uma novidade utilizada por poucas empresas de CE, para se tornar uma ferramenta de negócios fundamental para o ramo, pois eles conseguem aprender sobre o consumidor e recomendar produtos que serão de seu interesse.

Os SR já têm sido largamente utilizados em sites de compra como: Amazon.com[5], pioneira no CE lançada em 1995 pelo empreendedor Jeff Bezos, e Submarino[6], que disponibilizam aos consumidores produtos diversificados como, eletrodomésticos, livros, informática e acessórios, beleza e saúde, dentre outros (KAJIMOTO et al., 2008).

Para fazer as recomendações é fundamental que os SR coletem informações sobre os usuários. A coleta de informação pode ser realizada de duas maneiras (REATEGUI; CAZELLA, 2005):

1) explicita: o usuário indica espontaneamente o que lhe é importante. Como exemplo, a loja Submarino disponibiliza uma seção ao consumidor denominada “Avalie este produto e incremente suas sugestões” com as opções seguintes de notas: 1 (Ruim), 2 (Regular),

[5] www.amazon.com
[6] www.submarino.com.br

3 (Bom), 4 (Ótimo) e 5 (Excelente), além das opções “Eu já tenho” e “Não tenho interesse neste produto”, conforme Figura 1.

2) implícita: através de ações do usuário, como compra e busca por produtos.

Figura 1 Método de coleta de informação explícita da loja Submarino
Fonte: www.submarino.com.br

2.2 Filtragem de Informação (FI) aplicada a SR

A FI é uma técnica utilizada para filtrar uma grande quantidade de informação e entregar a determinada pessoa só a informação que lhe é relevante. A técnica de FI é comumente utilizada pelos SR para conseguir fazer recomendações de produtos para consumidores de lojas de CE. Os métodos de

FI utilizados pelos SR são mostrados na Figura 2, e explicadas nas seções seguintes.

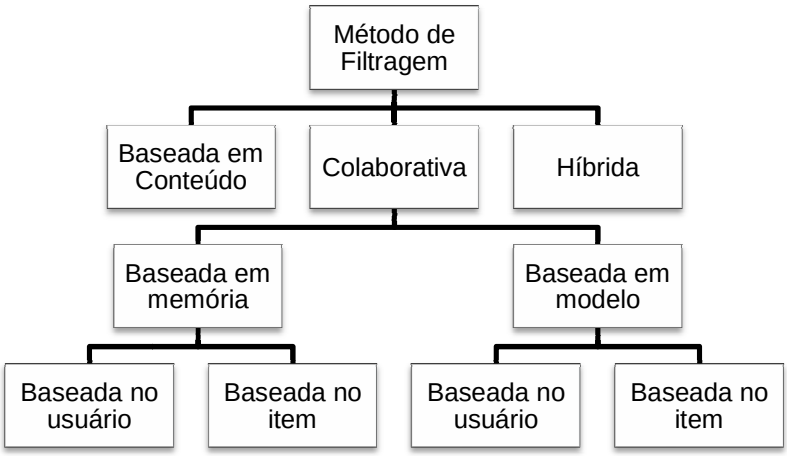

Figura 2 Classificação dos sistemas de Filtragem de Informação

2.2.1 Filtragem Baseada em Conteúdo (FBC)

A Filtragem Baseada em Conteúdo constrói um perfil de usuário baseado nas características dos itens na qual o usuário teve alguma interação, seja de maneira explícita ou implícita. Para recomendar um item, a FBC faz a correlação entre o conteúdo deste item e os interesses do usuário extraídos de seu perfil.

Nos métodos de recomendação que utilizam a FBC a utilidade de um item para um usuário é estimada baseada nas utilidades já assinaladas por ele a outros itens similares (ADOMAVICIUS; TUZHILIN, 2005, apud BERNARTT, 2008). É preciso entender as características dos itens que o usuário já avaliou com notas altas no passado, ou comprou, assim, itens que possuírem grande grau de semelhança com estes serão recomendados.

Exemplos de tecnologias aplicadas para FBC são classificadores bayesianos (MOONEY; BENNETT; ROY, 1998, apud BERNARTT, 2008) e técnicas de aprendizado de máquinas, incluindo agrupamento, árvores de decisão e Redes Neurais Artificiais (RNA) (PAZZANI; BILLSUS, 1997 apud BERNARTT, 2008).

A FBC possui problemas associados à super-especialização, pois não possui métodos que permitam recomendações de itens diferentes dos que o usuário já tenha visto antes (e indicou ter gostado). Como exemplo, se um usuário comprou somente filmes de terror, então a FBC irá somente lhe recomendar filmes de terror. Outro problema da FBC é não conseguir filtrar itens baseado em atributos subjetivos, como a qualidade de um filme.

Exemplo de utilização da técnica de FBC é a proposta de Gazzanal e Silveira (2009) de desenvolver um protótipo de SR, denominado RecomenTur, para a área de turismo. Através da técnica de FBC e informações coletadas de forma explícita e implícita o RecomenTur sugere ao usuário pacotes turísticos que se encaixem no seu perfil. Outro exemplo é o SR para Bibliotecas digitais sob perspectiva da Web Semântica, cujo objetivo é combinar informações do usuário obtidas a partir do seu currículo *Lattes* com informações referentes aos artigos para gerar a recomendação personalizada (LOPES; SOUTO; OLIVEIRA, 2006).

2.2.2 Filtragem Colaborativa (FC)

O objetivo da FC é recomendar novos itens ou predizer a utilidade de um item para um determinado usuário, utilizando como base os dados dos usuários similares à ele, ou seja que mostraram os mesmos interesses que ele, comprando ou avaliando com notas altas os mesmos.

As predições correspondem aos mesmos valores das avaliações dadas aos itens pelos usuários, como valores entre 1 a 5. Já as recomendações podem ser feitas através de uma lista com N itens que o usuário poderá gostar. Esta lista não deve conter itens já comprados pelo usuário ativo (usuário no qual se pretende fazer recomendações).

Existem três classes de FC que são: FC baseada em memória, FC baseada em modelo e FC Híbrida (FCH) apresentadas a seguir.

2.2.2.1 Filtragem Colaborativa Baseada em Memória

A filtragem colaborativa baseada em memória utiliza toda a base de dados que contém a relação entre usuários (u) e item (i) para fazer predições ou recomendações. Uma matriz usuário-item R de dimensão n x m pode ser utilizada para representar a relação dos n usuários sobre os m produtos. Cada célula, $r_{i,j}$ da matriz R pode conter: valores de avaliações, ou os valores zero ou um. O valor um significa que o i-ésimo consumidor comprou o j-ésimo item, e o valor zero significa que o o i-ésimo consumidor não comprou o j-ésimo item. A FC baseada em memória pode ser dividida em baseada no usuário e baseada no item.

A baseada no usuário busca encontrar vizinhos, ou seja, um conjunto de usuários que possuam gostos similares ao usuário ativo, por exemplo, tendem a comprar itens similares. Para cálculo da similaridade são atribuídos pesos a todos os usuários indicando seu grau de similaridade com o usuário ativo. As medidas de similaridade mais conhecidas são:

a. Coeficiente de correlação de Pearson cuja fórmula é:

$$sim(a,b) = \frac{\sum_i (r_{a,i} - \bar{r}_a)(r_{b,i} - \bar{r}_b)}{\sqrt{\sum_i (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_j (r_{b,i} - \bar{r}_b)^2}} \qquad (1)$$

Onde $sim(a,b)$ representa a similaridade entre os usuários *a e b*; $r_{a,i}$ e $r_{b,i}$ são as avaliações dos usuários a e b para o item i respectivamente, e $\bar{r}_a$ e $\bar{r}_b$ são os valores médios de avaliações do usuário a e b respectivamente.

b. Método do cosseno: dois usuários são tratados como um vetor e a similaridade é medida através do cálculo do cosseno entre esses vetores, conforme fórmula (2).

$$\cos(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|_2 * \|\vec{b}\|_2} \qquad (2)$$

Calculada a similaridade, os vizinhos são encontrados através de métodos como *Center-based* ou Vizinhança agregada (SARWAR et al., 2000). Depois que os vizinhos são formados, os sistemas utilizam algoritmos diferentes capazes de combinar suas preferências e assim fazer uma predição ou recomendação para o usuário ativo (SARWAR et al., 2001).

A baseada no item calcula a similaridade entre os itens, podendo ser utilizados tanto o método do cosseno (neste caso os itens são tratados como vetores) como o coeficiente de correlação de Pearson, conforme equação (3). Onde U é o conjunto de usuários que avaliaram ambos os itens i e j, $r_{u,i}$ representa a avaliação do usuário u sobre o item i e $\bar{r}_i$ é a média de avaliação para o i-ésimo item.

$$sim(i,j) = \frac{\sum_{u\in U}(r_{u,i}-\bar{r}_i)(r_{u,j}-\bar{r}_j)}{\sqrt{\sum_{u\in U}(r_{u,i}-\bar{r}_i)^2}\sqrt{\sum_j(r_{u,j}-\bar{r}_j)^2}} \qquad (3)$$

A FC baseada em memória possui algumas limitações como o problema da esparsidade, pois diante da grande quantidade de produtos disponíveis principalmente em grandes lojas de CE como a Amazon.com, usuários ativos compram ou avaliam apenas 1% dos produtos disponíveis. Logo, a matriz usuário-item que representa as transações dos clientes, possuirá muitas células

vazias, dificultando assim fazer as associações entre os usuários ou itens e as recomendações produzidas não serão tão precisas. Outra limitação é a escalabilidade, pois algoritmos baseados no usuário requerem a computação de milhares de produtos e clientes, que crescem o tempo todo. Assim, ao ter que responder às solicitações de milhares de usuários ao mesmo tempo um SR pode sofrer sérios problemas de escalabilidade.

2.2.2.2 Filtragem Colaborativa Baseada em Modelo

Na FC baseada em modelo, os dados sobre os usuários e os itens são utilizados para criar um modelo, que é então usado para fazer predições ou recomendações de itens. Segundo Sarwar et al. (2001) para construção do modelo podem ser utilizados algoritmos de aprendizagem de máquina, como: redes bayesianas, que formulam um modelo probabilístico para a FC; clusterização que trata a FC como um problema de classificação; métodos baseados em regras, que aplicam algoritmos que descobrem regras de associação entre itens previamente adquiridos e gera a recomendação baseado na força da associação entre os itens.

Assim como a FC baseada em memória, a FC baseada em modelo também pode ser divida em baseada no usuário e baseada no item.

A baseada no usuário constrói um modelo com base no usuário, e para isso pode utilizar diferentes técnicas como clusterização, onde usuários similares são agrupados em um mesmo *cluster*. Este modelo é então utilizado para estimar a probabilidade de que o usuário alvo pertença a determinado *cluster* C, que pode então ser usado para fazer predições de avaliações para o usuário alvo.

Na baseada no item a construção do modelo é com base nos itens, utilizando medidas de similaridade como a Probabilidade Condicional (Deshpande e Karypis, 2004) que pode ser expressa pela equação (4).

$$P(j|i) = Freq(ij)/Freq(i) \quad (4)$$

Onde $P(j|i)$ é a probabilidade condicional de se comprar o item *j* dado que o produto i já tenha sido comprado e $Freq(X)$ representa o número de consumidores que compraram os itens no conjunto X. Esta probabilidade corresponde ao número de consumidores que compraram ambos os itens i e j dividido pelo número total de usuários que compraram i. O resultado da construção deste modelo pode ser expresso através de uma matriz de similaridade S, que contém os resultados da probabilidade condicional, conforme mostra a Figura 3.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
i_1	1	0.1	0	0.3	0.2	0.4	0	0.1
i_2	0.1	1	0.8	0.9	0	0.2	0.1	0
i_3	0	0.8	1	0	0.4	0.1	0.3	0.5
i_4	0.3	0.9	0	1	0	0.3	0	0.1
i_5	0.2	0	0.4	0	1	0.1	0	0
i_6	0.4	0.2	0.1	0.3	0.1	1	0	0.1
i_7	0	0.1	0.3	0	0	0	1	0
i_8	0.1	0	0.5	0.1	0	0.1	0	1

Figura 3 Matriz de similaridade entre itens *S*
Fonte: Adaptado de Hahsler (2010)

Utilizando esta matriz a recomendação de uma lista contendo N itens pode ser feita ao usuário ativo através do algoritmo de recomendação *Top-N* baseado em item proposto por Karypis (2001), conforme exemplificado a seguir.

Considerando que o conjunto de itens comprados pelo usuário ativo (u_a) seja representado por $U = \{i_1, i_5, i_8\}$, primeiramente o algoritmo identifica um

conjunto C de itens candidatos a serem recomendados unindo os K itens mais similares $\{i_1, i_2,...,i_k\}$ para cada item $i \in U$. Desta união devem ser removidos os itens que já foram comprados pelo usuário ativo. Considerando K = 3 e utilizando os dados da matriz S da Figura 3 o conjunto C será formado pelos itens $C = \{i_3, i_4, i_6\}$. Agora a similaridade entre cada item $c \in C$ e o conjunto U é calculada através da soma das similaridades entre todos os itens $i \in U$ e c, usando somente os k itens mais similares de j. O resultado deste cálculo da similaridade é mostrado na figura 4, onde os valores em negrito correspondem aos 3 itens mais similares aos itens $i \in U$, estes que por sua vez correspondem às linhas destacadas.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	
i_1	1	0.1	0	**0.3**	**0.2**	**0.4**	0	0.1	$K = 3$
i_2	0.1	1	0.8	0.9	0	0.2	0.1	0	$u_a = \{i_1, i_5, i_8\}$
i_3	0	0.8	1	0	0.4	0.1	0.3	0.5	
i_4	0.3	0.9	0	1	0	0.3	0	0.1	
i_5	**0.2**	0	**0.4**	0	1	**0.1**	0	0	
i_6	0.4	0.2	0.1	0.3	0.1	1	0	0.1	
i_7	0	0.1	0.3	0	0	0	1	0	
i_8	**0.1**	0	**0.5**	**0.1**	0	0.1	0	1	
			0.9	0.4		0.5		$\rightarrow$	Soma das similaridades

Figura 4 Soma das similaridades entre todos os itens $i \in U$ e c
Fonte: Adaptado de Hahsler (2010)

Por fim, os itens em C são classificados em ordem decrescente em relação à similaridade, e os primeiros N itens são selecionados como o conjunto de itens *Top-N* a serem recomendados. Neste exemplo, os $N = 2$ itens recomendados seriam: i_6 e i_3.

A FC baseada em modelo consegue superar o problema da escalabilidade presente na FC baseada em memória, pois quando uma recomendação é requisitada, as relações de similaridade entre os usuários ou entre os itens podem ser encontradas através do modelo previamente construído e não calculada em tempo real, como ocorre na FC baseada em memória.

2.2.2.3 Filtragem Colaborativa Híbrida

Algoritmos que utilizam ambas as FC, baseadas em memória e baseadas em modelo, denominada Filtragem Colaborativa Híbrida (FCH) podem ser encontrados em Pennock et al. (2000) e Xue et al. (2005). A combinação consegue superar o problema da esparsidade, conforme mostrado em Xue et al. (2005) aumentando a eficiência e precisão das recomendações.

Ambas as técnicas, FC baseada em memória e FC baseada em modelo, superam algumas das limitações da FBC como: (i) é possível recomendar itens ao usuário que são muito diferentes daqueles que ele já mostrou preferência antes e (ii) as recomendações são baseadas na qualidade ao invés de propriedades objetivas dos próprios itens (SHARDANAND; MAES, 1995). Porém SR que utilizam algoritmos de FC baseada em memória e SR que utilizam algoritmos de FC baseada em modelo possuem um problema denominado *Cold-Start* apresentado a seguir.

2.2.3 Filtragem Híbrida (FH)

A FH combina a FBC com a FC, assim ajuda a contornar certas limitações existentes nestes dois métodos de filtragem. As formas possíveis de se combinar a FBC com a FC são:

- Ponderada (ou *weighted* em inglês): os SR implementam a FC e FBC separadamente e são associados pontos ou votos às técnicas de acordo com seus resultados. Como exemplo, pode-se iniciar atribuindo pesos iguais para ambas as técnicas de filtragem e conforme as predições sobre as avaliações dos usuários são confirmadas ou não esses pesos vão sendo ajustados.
- Mista: recomendações geradas pelas duas técnicas (FC e FBC) são combinadas no processo final de recomendação (BURKE, 2002).
- Combinação seqüencial: os perfis dos usuários são construídos pela FBC baseada nos itens acessados por eles, e então a FC é aplicada para fazer predições baseada no perfil dos usuários. (KIM et al., 2006, apud ALBADVI; SHAHBAZI, 2009).
- Comutação: o sistema utiliza algum critério para comutar entre a FBC e FC. Como exemplo, pode ser aplicada a FBC primeiro e caso a recomendação não seja de confiança então se utiliza a FC (BURKE, 2002).

3 PROBLEMA *COLD-START*

O problema *Cold-Start* (freqüentemente conhecido como o Problema da Inicialização) ocorre quando os dados (avaliações, compras, buscas, etc.) sobre os itens ou usuários não estão disponíveis no sistema, ou são muito escassos, e desta forma não é possível fazer recomendações (PARK; CHU, 2009). Existem dois tipos do problema *Cold-Start*, denominados *Cold-Start User* e *Cold-Start Item* descritos a seguir.

É possível observar um exemplo do problema *Cold-Start User* através da situação seguinte: a partir de uma matriz que armazena dados de compras dos usuários, conforme Figura 5 (a), a FC consegue identificar que ambos os usuários u_1 e u_3 compraram os itens i_2, i_3, i_4 e i_5, logo esses usuários possuem preferências comuns. Através desse conhecimento obtido, é possível recomendar o item i_1 para o usuário u_3, pois este item já foi comprado por um usuário com gostos semelhantes ao dele (usuário u_1). Porém, dado que o usuário u_5 não comprou nenhum item, Figura 5 (b), a FC não consegue encontrar relações entre este e os demais usuários, ou seja, usuários com compras em comum para fazer recomendações. Neste último caso, ocorreu o problema *Cold-Start User*.

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	1	1
u_2	0	0	1	0	1
u_3	0	1	1	1	1
u_4	0	1	0	1	0
u_5	0	0	0	0	0

(a)

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	1	1
u_2	0	0	1	0	1
u_3	0	1	1	1	1
u_4	0	1	0	1	0
u_5	**0**	**0**	**0**	**0**	**0**

(b)

Figura 5 (a) relação entre os usuários u_1 e u_3 e (b) exemplo do problema *Cold-Start User*

O problema *Cold-Start Item* (Problema do Novo Item) ocorre quando poucos ou nenhum usuário avaliou, comprou ou visualizou determinado item. A Figura 6 (a) mostra que dada um matriz que armazena dados de compras dos usuários a FC conseguiu encontrar uma relação entre os usuários u_1 e u_3, pois ambos compraram os itens i_2, i_3, i_4 e i_5. Esta relação indica que os usuários u_1 e u_3 usuários têm gostos parecidos, e que a compra de um dos usuários pode também agradar ao outro usuário. Logo é possível recomendar o item i_1, comprado pelo usuário u_1, para o usuário u_3 que ainda não comprou este item. Porém, se o item i_1 nunca tivesse sido comprado, como ilustra a Figura 6 (b), não seria possível associar este item aos demais itens comprados pelo usuário u_1 e assim fazer recomendações para usuários similares a ele. É possível dizer então que o item i_1 sofre do problema *Cold-Start Item*.

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	1	1
u_2	0	0	1	0	1
u_3	0	1	1	1	1
u_4	1	0	0	0	0
u_5	1	1	0	1	1

(a)

	i_1	i_2	i_3	i_4	i_5
u_1	**0**	1	1	1	1
u_2	**0**	0	1	0	1
u_3	**0**	1	1	1	1
u_4	**0**	0	0	0	0
u_5	**0**	1	0	1	1

(b)

Figura 6 (a) Relação entre os usuários $\boldsymbol{u_1}$ e $\boldsymbol{u_3}$ e (b) exemplo do problema *Cold-Start Item*

Um segundo exemplo, conforme Figura 7 (a), mostra que a FC conseguiu identificar a relação entre os itens i_2 e i_3, pois os usuários u_1, u_2, u_3 e u_5 sempre que compraram o item i_2 compraram o item i_5 também. Porém se o item i_2 nunca tivesse sido comprado a FC não conseguiria encontrar essa relação do item i_2 com o item i_5, conforme mostra a Figura 7 (b), nem mesmo com os

demais itens disponível na matriz. Esse exemplo mostra que o item i_2 sofre do problema *Cold-Start Item*.

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	1	1
u_2	0	1	1	0	1
u_3	0	1	1	1	1
u_4	1	0	0	0	0
u_5	1	1	1	1	1

(a)

	i_1	i_2	i_3	i_4	i_5
u_1	1	**0**	1	1	1
u_2	0	**0**	1	0	1
u_3	0	**0**	1	1	1
u_4	1	**0**	0	0	0
u_5	1	**0**	0	1	1

(b)

Figura 7 (a) Relação entre os itens $\boldsymbol{i_2}$ e $\boldsymbol{i_3}$ e (b) exemplo do problema *Cold-Start Item*

Quando comparados à FC, a FBC ou a FH apresentam um desempenho um pouco melhor diante do problema *Cold-Start*. Na FBC, mesmo para um produto que não possui nenhum dado de compra ou avaliações, bastando que suas características sejam similares as características dos outros produtos já comprados pelo usuário ativo este item será recomendado e não sofrerá do problema *Cold-Start Item*. E mesmo quando o usuário é novo no sistema e possui poucos dados de compra, para a FBC a recomendação não dependerá do relacionamento inicial deste usuário e os demais usuários. Bastando saber o conteúdo de um item comprado pelo usuário ativo já é possível fazer recomendações, e assim este usuário não sofrerá do problema *Cold-Start User*.

Já a FH por combinar ambas as filtragens, FBC e FC, conseguirá superar as limitações entre as duas e minimizar ou solucionar o problema *Cold-Start*.

A quantidade de dados sobre itens ou usuários que deve ser atingida para que os SR superem o problema *Cold-Start* e gerem recomendações relevantes depende de fatores como as características dos itens e usuários. Como exemplo, um novo item que seja procurado somente por usuários com gostos muito

peculiares terá na base de dados pouco histórico de avaliações, compras ou buscas, porém, mesmo diante de poucos dados a FC baseada no usuário ou FC baseada no item conseguirão recomendar de forma relevante este item para um grupo específico de usuários. Diferentemente, um item muito procurado, como um lançamento popular, por possuir uma quantidade muito grande de dados facilmente será relacionado a vários outros itens, mesmo que não sejam relevantes para serem recomendados ao usuário.

4 METODOLOGIA

4.1 Classificação da Pesquisa

Quanto a sua natureza e objetivos este trabalho é classificado como pesquisa aplicada e exploratória respectivamente, pois se buscou fazer uma análise de algoritmos de FI para solução do problema *Cold-Start Item*, no qual um deles será aplicado em um SR desenvolvido pela empresa Chaordic Systems.

De acordo com a abordagem, este trabalho utiliza métodos de pesquisa qualitativa, pois buscou entender sobre a técnica de FI, descrever soluções para o problema *Cold-Start Item* que utilizam a FI, para assim fazer uma análise destas soluções encontradas.

4.2 Critérios para seleção dos algoritmos de FI

Para seleção dos algoritmos de FI encontrados algumas das características dos artigos foram levadas em consideração, como: ano de publicação; quantidade de citações de acordo com o Google Acadêmico[7]; dados utilizados para teste; métrica de avaliação e os resultados obtidos. Além disso, buscou-se selecionar algoritmos que utilizem as características dos itens no processo de recomendação, o que ajuda na recomendação de itens que sofrem do problema *Cold-Start Item*.

[7] http://scholar.google.com.br

5 RESULTADOS E DISCUSSÃO

Atualmente os algoritmos de FI implementados pela empresa Chaordic Systems não conseguem solucionar o problema *Cold-Start Item*. Pois, dada uma matriz que armazena dados dos usuários e itens, os algoritmos buscam identificar quais itens possuem relação com um item já comprado ou avaliado pelo usuário ativo para assim recomendá-los. Porém, aqueles itens que possuem pouco ou nenhum dado de compra ou avaliação possuem poucas chances de serem recomendados.

Logo, algoritmos possíveis de solucionar este problema seriam aqueles que conseguem exprimir as relações de similaridade entre os itens com base em suas características, como no caso de filmes, tais características seriam atores, diretores ou gênero. Assim, um item mesmo sem nenhum dado de compra ou avaliação, mas com suas características conhecidas, pode ser recomendado aos usuários.

Assim, através de um levantamento de algoritmos de FI mais relevantes da última década para solucionar o problema *Cold-Start Item*, foi possível selecionar os oito algoritmos descritos na seção 5.1.

5.1 Descrições dos Algoritmos de FI selecionados

A seguir são descritos os algoritmos de FI que foram selecionados como sendo possíveis soluções para o problema *Cold-Start Item*.

Algoritmo de Christakou e Stafylopatis

Os autores Christakou e Stafylopatis (2005) propõem combinar os resultados das técnicas de FBC e FC para construir um SR Híbrido que fornece recomendações mais precisas sobre filmes. O método utiliza o conteúdo dos filmes (gênero, sinopse e elenco) que os indivíduos já avaliaram; os filmes que usuários com preferências similares tenham gostado, e a opinião dos outros usuários, para que não fique restrito a recomendar somente filmes similares ao que o usuário já tem preferência.

A parte de FBC foi construída através da criação de três RNAs (Perceptron de múltiplas camadas) para cada usuário, que corresponde às três características dos filmes: gênero, sinopse e elenco, no qual o usuário já avaliou. Durante o treinamento as entradas da rede correspondem ao gênero, sinopse e elenco do filme. Como exemplo, para o treinamento da uma rede neural Gênero, as possíveis entradas são: suspense, terror, romance, cujo valor é 1 quando esta entrada está presente no filme. A saída de cada rede é de tamanho cinco, simulando a escala de 5 graus na qual o usuário avalia um filme. Este valor é transformado em valores binários, zero indicando que o filme não será recomendado e um se o filme for recomendado. O resultado de cada rede neural é salvo em uma matriz, obtendo-se três matrizes para cada usuário que representam o resultado da parte correspondente à FBC.

A parte de FC utiliza o coeficiente de correlação de Pearson (Equação 1) para encontrar a correlação entre um usuário específico e o restante dos usuários.

A opinião do usuário y é levada em consideração em duas situações: quando sua correlação com o usuário x é muito grande ($r > 0.4$) e ele forneceu notas altas para os filmes (4 ou 5), neste caso o contador de propostas positivas é incrementado. Quando o usuário y possui uma correlação muito baixa ($r < -0.5$)

com o usuário y, considera-se que as preferências entre os usuários não combinam, logo um contador de propostas negativas é incrementado quando o usuário y forneceu notas muito baixas aos filmes. Estes contadores fornecem a porcentagem pelo qual cada filme é recomendado e são utilizados na fase de combinação entre a FBC e FC. Se seu valor é maior que 50% o filme é recomendado caso contrário não.

Na combinação entre a FBC e a FC, é preciso escolher 5 filmes de interesse do usuário a serem recomendados. Para recomendar estes filmes são seguidas as seguintes etapas:

1. Selecione os filmes que são sugeridos por todos os quatro critérios – tipo, participantes, sinopse (critério por conteúdo) e opinião de outros usuários (critério colaborativo).
2. Adicione os filmes que satisfazem o critério colaborativo usando um alto limiar.
3. Adicione os filmes que são sugeridos por todos os critérios por conteúdo (especialmente no caso de novos filmes, para os quais não estão disponíveis avaliações).
4. Adicione os filmes que são sugeridos por exatamente dois critérios por conteúdo, mais o critério de colaboração.
5. Adicione os filmes que são sugeridos por dois critérios por conteúdo.
6. Adicione os filmes que são sugeridos por um critério por conteúdo.
7. Finalmente, se ainda faltar filmes a serem recomendados, sugiram os filmes mais populares.

O algoritmo de Kim e Li

O algoritmo de FC baseado em modelo (baseado no item) proposto pelos autores Kim e Li (2004) utilizam os atributos dos itens na FC para complementar as avaliações dos usuários e assim conseguir extrair similaridade entre os itens.

Através do algoritmo de clusterização *K-means* itens são associados a grupos de acordo com seus atributos (como exemplo atores, diretor, gênero e sinopse de um filme). Feito isso, é calculada a probabilidade de cada item pertencer a cada grupo, e assim a matriz item-usuário é estendida tal que os grupos formados são registrados como novos usuários, conforme exemplo mostrado na Tabela 1.

Tabela 1 Matriz item-usuário estendida.

	Jack	Oliver	Peter	Grupo 1	Grupo 2
Item 1	5		1	98%	4%
Item 2		4		96%	5%
Item 3				98%	4%

Fonte: Adaptado de Kim e Li (2004)

Esta nova matriz item-usuário é então utilizada para construir uma classe *z*, tratada como uma comunidade de item, que agrupa itens similares. O algoritmo de clusterização *k-Medoids* (HAN; KAMBER, 2000) cria a classe *z* da seguinte forma:

1. *K* itens são selecionados aleatoriamente como o centro dos *clusters*.
2. Através do coeficiente de correlação de Pearson (Equação 3), a similaridade de um item com o item presente no centro do cluster é calculada. O item é então atribuído ao melhor *cluster*, ou seja, cuja sua similaridade com o item do centro do *cluster* é maior. O novo

centro será o item que possuir a melhor correlação com todos os outros itens no *cluster*.

3. O passo 2 deve ser repetido até que não existam itens a serem associados a *clusters*.
4. A cada *cluster* é associado um representante, que será o item onde o voto do usuário, no qual se pretende fazer a predição, é a média das avaliações feitas por este usuário aos itens membros daquela comunidade de item.

Para calcular a predição do voto do usuário assume-se que as avaliações para certo item na comunidade z satisfaz a distribuição Gaussiana. A predição é obtida através das equações 5 e 6:

$$p(z|y) = \frac{1/ED(V_y,V_z)}{\sum_{z\prime=1}^{k} 1/ED(V_{y,}V_{z\prime})} \qquad (5)$$

$$\mu_{u,z} = \frac{\sum_{y\in U_z} v_{u,y} p(z|y)}{\sum_{y\in U_z} p(z|y)} \qquad (6)$$

Onde $p(z|y)$ é o grau de pertinência do item y à classe z, $\mu_{u,z}$ é a avaliação média do usuário u na classe z, V_y corresponde ao vetor de avaliações do item y, V_z é o vetor do item representativo da classe z, $ED(.)$ é distância euclidiana entre dois vetores, $v_{u,y}$ é o voto do usuário u sobre o item y e U_z é o domínio dos itens que pertencem a comunidade de item z.

Os algoritmos de Han e Karypis

Han e Karypis (2005) apresentam dois algoritmos de FC baseados em modelo (baseado no item) que recomendam itens de acordo com suas características. Tais algoritmos superam a limitação do algoritmo de recomendação *Top-N* baseado em item proposto por Karypis (2001) e

apresentado na seção 2.2.2.2, que não consegue medir a similaridade entre novos itens com itens existentes.

O algoritmo utiliza três conjuntos de dados para treinamento: uma cesta de dados de treinamento (ou *training basket data* em inglês) correspondente ao histórico de vendas passado, na qual cada uma corresponde a uma única transação onde um conjunto de produtos foi vendido a um cliente; um catálogo de produtos de treinamento (ou *training product catalog* em inglês) contendo todos os produtos em uma cesta de dados de treinamento e suas características, e também um catálogo de produtos atual que contém produtos disponíveis atualmente (incluindo novos produtos que não estão no catálogo de produtos de treinamento) e suas características.

Primeiramente, utilizando o algoritmo de recomendação *Top-N* baseado em item (KARYPIS, 2001) é criado um modelo de recomendação M. Após isso, os passos do primeiro algoritmo proposto são:

1. Dado um conjunto de produtos X em uma cesta de compras, encontre produtos similares S à X usando as características dos produtos de X a partir do catálogo de produtos de treinamento.
2. Utilizando os produtos do conjunto S, encontre produtos recomendáveis R utilizando o modelo de recomendação M.
3. Atribua uma pontuação de recomendação ou confiança às características dos produtos em R. A pontuação de uma característica será a soma da pontuação de recomendação de cada produto que possui esta característica. Portanto, características que estão presentes em muitos produtos recomendados e também em produtos altamente recomendáveis possuirão uma pontuação maior.
4. Cada produto presente no catálogo atual receberá uma pontuação através da soma das pontuações das características encontradas no

passo 3 que estiverem presentes neste produto. Assim os N produtos com maiores pontuações serão recomendados.

O segundo método proposto utiliza diretamente as características dos itens que estão em um cesto para recomendação. Neste método, dado uma cesta de dados de treinamento, são construídas regras de associação RA da seguinte forma: para cada transação (conjunto de produtos vendidos a um cliente) contida na cesta de dados de treinamento e para cada produto na transação, é construída uma regra de associação onde este produto é uma conseqüência da regra e todos os outros são os antecedentes da regra. Como exemplo, dada uma transação $\{a, b, c\}$, as regras de associação serão: $\{a, b\} \rightarrow c$, $\{a, c\} \rightarrow b$, $\{b, c\} \rightarrow a$. Os passos para recomendação podem ser resumidos da seguinte forma:

1. Dado um conjunto de produtos X em uma cesta de compras, encontre *matching rules*[8] R de X combinando as características dos produtos de X com as características dos produtos no lado antecedente das regras em RA.
2. Encontre características recomendadas F agregando as características dos produtos na conseqüência das regras em R. Nesta etapa similar à etapa 3 do primeiro algoritmo, as características recebem pontuações.
3. Encontre *Top-N* produtos adequados utilizando F do catálogo de produtos atual. Esta etapa é igual à etapa 4 do primeiro algoritmo.

[8] *Matching rules* determinam se valores de atributos são logicamente iguais uns aos outros.

O algoritmo de Hofmann

Hofmann (2004) propõe um algoritmo de FC baseado em modelo cujo domínio consiste em conjunto de usuários $u = \{u_i, \dots, u_n\}$, um conjunto de itens $y = \{y_1, \dots, y_m\}$ e um conjunto de avaliações v. São considerados dois tipos de problema de predição. O primeiro, chamado "predição forçada", busca predizer a preferência para um item dada a identidade do usuário, ou seja, aprender o mapeamento $g: U \times Y \rightarrow V$. Ou também, está interessado na probabilidade condicional $P(v|u, y)$ que o usuário u irá avaliar o item y com v. Baseado nesta probabilidade pode ser definida uma função de predição determinística $g(u, y) = argmax_v P(v|u, y)$. O segundo problema denominado predição livre tem como objetivo aprender probabilidades $P(v, y|u)$ para predizer ambos, o item selecionado y e a avaliação associada v.

Este algoritmo é generalização da técnica estatística denominada probabilistic Latent Semantic Analysis (pLSA) (Hofmann, 2001). Neste algoritmo uma variável escondida Z com estado z é introduzida para cada par usuário-item (u, y). O estado z modela uma causa escondida: determinada pessoa u selecionou o item y "por causa" de z, ou seja, o estado z fornece uma explicação hipotética para uma avaliação fornecida pelo usuário que diretamente não poderia se observada. A variável Z representa clusters de usuários associado a cada par usuário-item.

O algoritmo introduz um parâmetro local $\mu_{y,z} \in \mathcal{R}$ e um parâmetro de escala $\sigma_{y,z} \in \mathcal{R}^+$ para cada comunidade z e cada item y e assim a probabilidade de avaliação v é dada pela equação (7):

$$P(v|y, z) = P\left(v; \mu_{y,z}, \sigma_{y,z}\right) = \frac{1}{\sqrt{2\pi}\sigma_{y,z}} exp\left[-\frac{\left(v-\mu_{y,z}\right)^2}{2\sigma^2{}_{y,z}}\right] \quad (7)$$

O algoritmo de Tiraweerakhajohn e Pinngern

O algoritmo proposto por Tiraweerakhajohn e Pinngern (2004), utiliza a técnica de mineração de dados denominada mineração de regras de associação, para encontrar relações de similaridades entre os atributos dos itens (parte de FBC), e assim encontrar itens similares (parte de FC).

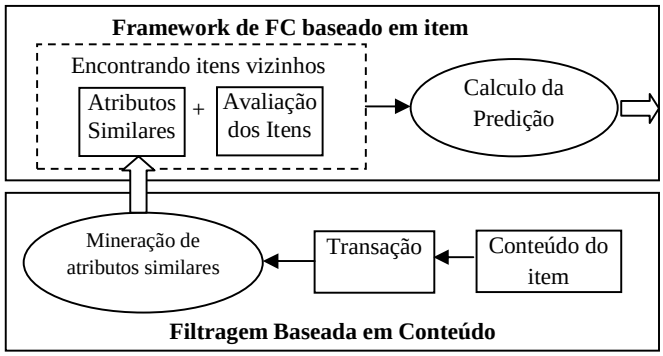

Figura 8 Framework de FC baseado na adição de item

Os passos do algoritmo, mostrados na Figura 8 são descritos a seguir:

1. Através de uma matriz represente os conteúdos dos itens como um conjunto de valores booleanos, onde cada linha da matriz representa um item e cada coluna representa o valor de um atributo. Então converta os conteúdos dos itens em transações de atributos.
2. Utilize o algoritmo de mineração de regras de associação Apriori (AGRAWAL; SRIKANT, 1994) para extrair regras de associação entre cada par de atributos nas transações de atributos, e também os valores de suporte (probabilidade que dois conjuntos de dados ocorram juntos em uma transação) e confiança (probabilidade de que certa regra seja válida) correspondentes, conforme Tabela 2.

Tabela 2 Exemplo de regras de associação.

Regra	Suporte	Confiança
Ação⇒Comédia	25%	34%
Ação⇒Drama	7%	11%
Ação⇒Romance	14%	17%
Comédia⇒Drama	50%	66%
Comédia⇒Romance	5%	7%

Fonte: Tiraweerakhajohn e Pinngern (2004)

Os valores de confiança são utilizados para preencher uma matriz de similaridade conforme Figura 9.

	Ação	Comédia	Drama	Romance
Ação	1	0.34	0.11	0.17
Comédia	–	1	0.66	0.7
Drama	–	–	1	–
Romance	–	–	–	1

Figura 9 Matriz de similaridade atributo-atributo
Fonte: Tiraweerakhajohn e Pinngern (2004)

3. Combine as similaridades entre os atributos dos itens conforme equações (8, 9 e 10) com a similaridade entre as avaliações dos itens através do método do cosseno descrito na seção 2.2.2.1. Assim é possível obter a similaridade total entre os itens conforme equação (11).

$$ISim(t,c,n) = \sum_{t_i \in t} \frac{ASim(t_i,c,n)}{t} \quad (8)$$

$$ASim(t_i,c,n) = 1, if \exists t_i \in t : t_i = c_j \quad (9)$$

$$= \frac{\sum_{j=1\ldots n} Sim(t_i,c_j)}{n} \quad (10)$$

$$CSim(i,j) = \alpha \times ISim(i,j) + (1-\alpha) \times RSim(i,j) \quad (11)$$

Onde α é um parâmetro de combinação especificando o peso da semelhança.

4. A predição de avaliações é feita através da fórmula (12).

$$P_{u,i} = \frac{\sum_{k \in k}(CSim(i,k) * R_{u,k})}{\sum_{k \in k}(|CSim(i,k)|)} \quad (12)$$

O algoritmo de Liu, Wang, Fang e Mi

Liu et al. (2007) propõem um algoritmo de FC baseada em modelo (baseada no item) que calcula a similaridade entre os itens através do mapeamento destes com respectivos documentos que descrevem seus conteúdos e características. Em seguida são feitas predições de acordo com estas similaridades entre os itens.

Como entrada o algoritmo utiliza uma matriz usuário-item R com avaliações dos usuários, e um conjunto de documentos M que descrevem os itens. A saída gerada é a predição P de avaliação do usuário. As etapas do algoritmo são:

1. Calcule a similaridade entre os documentos através do da medida estatística *Term Frequency–Inverse Document Frequency* (TF-IDF) que indica a importância de uma palavra em um documento. O resultado deste cálculo deve ser armazenado em uma matriz item-item que expressa a relação existente entre dois itens, conforme Figura 10.

$$ISim = \begin{bmatrix} S_{1,1} & S_{2,1} & \cdots & S_{1,n} \\ S_{1,2} & S_{2,2} & \cdots & S_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ S_{1,n} & \cdots & \cdots & S_{n,n} \end{bmatrix}$$

Figura 10 Matriz de similaridade entre itens
Fonte: Liu et al. (2007)

2. Calcule a predição de itens ainda não avaliados para o usuário ativo combinando a matriz de similaridade entre itens (*ISim*) de acordo com a equação (13):

$$R_{a,i} = \frac{\sum_{j=1}^{m} ISim_{i,j}(R_{a,j})}{ISim_{i,j}} \qquad (13)$$

Onde $R_{a,i}$ representa a predição de avaliação do usuário ativo para os itens ainda não avaliados, $ISim_{i,j}$ significa a semelhança entre os itens i e j. Os resultados desta etapa são adicionados à matriz usuário-item R gerando a nova matriz R'.

3. Encontre uma matriz de similaridade entre usuários (Figura 11), utilizando o coeficiente de correlação de Pearson (descrito na seção 2.2.2.1) na matriz R'.

$$USim = \begin{bmatrix} S_{1,1} & S_{2,1} & \cdots & S_{1,n} \\ S_{1,2} & S_{2,2} & \cdots & S_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ S_{1,n} & \cdots & \cdots & S_{n,n} \end{bmatrix}$$

Figura 11 Matriz de similaridade entre usuários
Fonte: Liu et al. (2007)

A predição de avaliações para itens ainda não avaliados pelo usuário ativo ocorre de acordo com a equação (14).

$$P_{a,i} = \bar{r} + \frac{\sum_{u=1}^{n}[(r_{u,i} - \bar{r}_u) * S_{a,u}]}{\sum_{u=1}^{n} S_{a,u}} \qquad (14)$$

Onde, $S_{a,u}$ representa a similaridade entre o usuário ativo e o usuário u presente na matriz de similaridade $USim$.

O algoritmo de Wang e Kong

Wang e Kong (2007) propõem a utilização de um método de recomendação baseado em FCH que utiliza as informações semânticas das

categorias dos itens, informações demográficas dos usuários, e as avaliações dos usuários presentes na matriz usuário-item.

Antes do estágio de recomendação são implementados dois algoritmos: um para construção de uma ontologia de categoria de item e outro para clusterização de usuários.

Para construção de uma ontologia de categoria de item primeiramente é preciso conhecer todas as categorias dos objetos que estão presentes no sistema. Em seguida é construída uma árvore com as categorias em hierarquia, onde cada unidade de categoria é a combinação das unidades de categoria de sua camada anterior, como mostra Figura 12. Usando uma matriz todos os itens são armazenados de acordo com suas categorias.

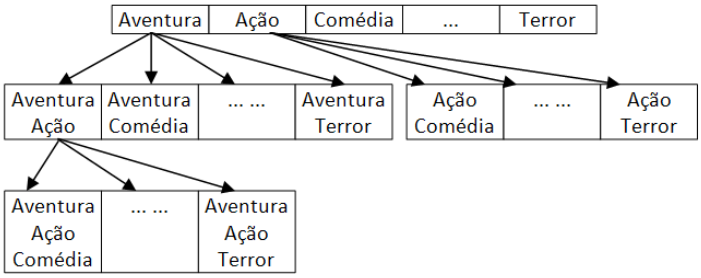

Figura 12 A hierarquia ontológica de categoria de objeto
Fonte: Wang e Kong (2007)

Assim, a similaridade entre dois itens t_i e t_j que compartilham n_{ij} categorias em comum em um total de N categorias para determinado domínio, é calculada através da equação (15).

$$SIM(t_i, t_j) = \frac{n_{ij}}{N} \qquad (15)$$

O algoritmo para clusterização dos usuários ocorre da seguinte forma:

1. Calcula-se a similaridade entre dois usuários, denominada $sim(u_i, u_j)$ através dos seus históricos de avaliações contidos na

matriz usuário-item, utilizando o coeficiente de correlação de Pearson (Equação 1).

2. Calcula-se a similaridade entre dois usuários utilizando Informações demográficas destes, como idade e profissão. A similaridade demográfica final entre dois usuários é calculada através média ponderada de todas as similaridades demográficas, através da equação (16):

$$sim(u_i, u_j)^2 = \sum_{k=1}^{n} {s_{ij}}^k * w_k \qquad (16)$$

Onde ${s_{ij}}^k$ é a *k*-ésima característica demográfica similar entre o usuário u_i e u_j, respectivamente; w_k é o maior fator correspondente a *k*-ésima característica demográfica no cálculo de similaridade, e n é o número de características demográficas selecionadas.

3. Calcule a similaridade de interesses e preferências de dois usuários baseado nas similaridades semânticas de itens que estes acessaram ou avaliaram, através da fórmula (17).

$$sim(u_j \rightarrow u_i) = \frac{\sum_{k2} \max_{k1}\left(SIM\left(t_{k2'}, t_{k1}\right)\right)}{k2} \qquad (17)$$

Onde o usuário u_i acessou/avaliou os itens t_1 a t_{k1} e o usuário u_j acessou/avaliou os itens $t_1{}'$a $t_{k2}{}'$.

4. Calcule a similaridade final entre pares de usuários utilizando a média ponderada das três similaridades calculadas acima:

$$sim(u_i, u_j) = \sum_{k=1}^{3} sim(u_i, u_j)^k * \alpha_k \quad (18)$$

$$\sum_{k=1}^{3} \alpha_k = 1 \qquad (19)$$

Onde α_ké a o fator de peso (*weightiness*) dado.

5. Utilizando o algoritmo de clusterização *K-means*, associe os usuários a *clusters*.

Para recomendação, primeiramente é construído um vetor de preferência para o usuário ativo $p_i = (c_{i1}, c_{i2,\ldots,} c_{i,n})$, onde $c_{i,j}$ representa o grau de interesse do usuário para a j-ésima categoria do objeto, e n representa a o número de categorias dos itens. Para cada usuário u_i, devem ser analisadas as categorias de cada item que ele acessou ou avaliou, e então ao seu vetor de preferências deve ser adicionado o valor 1 ao elemento correspondente de acordo com as categorias do item. O vetor é então organizado em ordem decrescente e os T*op-N* maiores elementos correspondem às categorias de item mais preferidas do usuário ativo.

Em seguida conhecido o *cluster* na qual o usuário ativo faz parte (criado no estagio de clusterização dos usuários), os usuários pertencentes ao mesmo grupo do usuário ativo irão predizer a pontuação de classificação (ou *the rating scores* do inglês) dos objetos que o usuário ativo nunca avaliou antes. Por fim, o sistema recomenda top-k itens para o usuário ativo.

O algoritmo de Leung, Chan e Chung

Leung, Chan e Chung (2008) apresentam o algoritmo Cross-Level Association RulEs (CLARE), com base no algoritmo Fuzzy Association Rule Mining and Multiple-level Similarity (FARAMS), desenvolvido pelos mesmos autores em 2006.

CLARE utiliza um modelo que compreende tanto relações entre usuário-item e item-item. Como mostrado na Figura 13 este modelo contém três camadas: usuário, item e atributos dos itens. A aresta que conecta o nó de um usuário e um nó de um item representa a avaliação do usuário para aquele item. Foi utilizada a técnica de mineração de regra de associação *fuzzy* sobre as avaliações dos usuários, para que estas ficassem compreendidas entre os valores

[0,1] indicando o grau pelo qual os usuários têm preferência pelos itens avaliados. Este modelo só captura a o grau de pertinência no qual o usuário gosta do item (valor 1). A aresta entre o nó item e nó atributo representa os atributos daquele item.

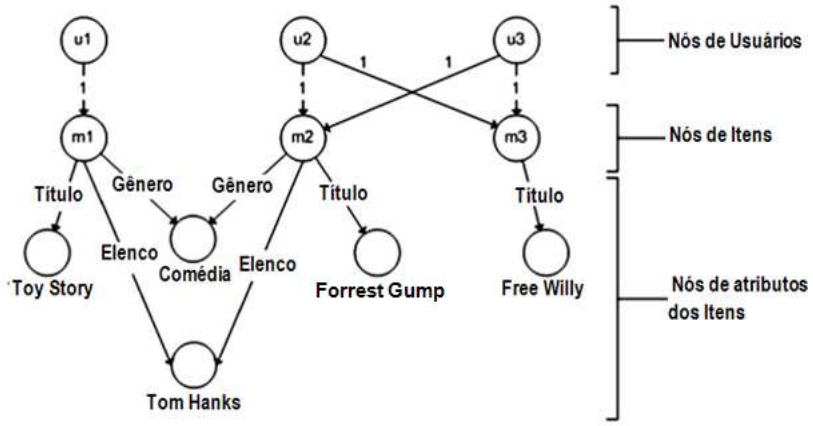

Figura 13 Modelo de preferência com relações usuário-item e item-item
Fonte: Adaptado de Leung, Chan e Chung (2008)

As etapas do algoritmo CLARE são:

Processamento dos dados

1. Utilizando a técnica de mineração de regra de associação *fuzzy* sobre as avaliações dos usuários dadas aos itens, estas avaliações vão ficar compreendidas entre os valores [0,1].
2. As preferências dos usuários para atributos dos itens devem ser calculadas. Como exemplo, utilizando o modelo da Figura 13 é possível verificar que o usuário tem preferência pelos atributos Toy Story, Tom Hanks e Comédia, pois ele tem preferência por itens que possuem estes atributos.

3. São geradas representações transacionais das preferências dos usuários tanto pelos itens como pelos atributos dos itens. Estas representações são feitas utilizando *vertical* TID-lists (ZAKI, 2000, apud LEUNG, CHAN E CHUNG, 2008), onde para cada item ou atributo é associada uma lista de transações onde ele ocorre. As representações transacionais das preferências do usuário pelos itens e pelos atributos são representadas como T_p e T_a, respectivamente.

Mineração de regras de associação

Dado um item denominado *targetItem*, utilizando o algoritmo de mineração descrito em Leung, Chan e Chung (2006) são criadas regras de associação para este item na forma $\{p\} \rightarrow targetItem$, que pode ser interpretado como "se o usuário gostou deste item p, então ele também gostará do item *targetItem*". Quando nenhuma regra de associação pode ser formada para determinado item, significa que este sofre do problema *Cold-Start Item*, logo os nós com seus atributos são utilizados para inferir preferências sobre ele. Utilizando como entrada o *targetItem*, a representação transacional das preferências dos usuário pelos itens (T_p), a representação transacional das preferências dos usuário pelos atributos dos itens (T_a) e a relação entre os nós dos itens e nós dos atributos dos itens denominada Domínio de Conhecimento (D), as etapas do algoritmo para encontrar as regras de associação são:

1. Dado o domínio de conhecimento (D) é recuperada a lista de atributos do *targetItem*, e a partir de T_a são recuperadas as *vertical* TID-lists desses atributos. As listas de atributos e suas *vertical* TID-lists são denominadas candidateAttr e T_{ac}, respectivamente.

2. Utilizando o algoritmo Apriori-gen *function* (AGRAWAL, IMIELINSKI e SWAMI, 1993, apud LEUNG, CHAN E CHUNG, 2008) são gerados *k*-conjuntos de itens freqüentes (targetAttr) a partir de candidateAttr, que correspondem a um conjunto de *k* itens que possuem um valor de suporte acima de um valor predefinido (minSupp). O suporte de uma regra "A→B" é a porcentagem de transações em uma base de transações (T) contendo A∩B. Para estes conjuntos de itens são retidos suas *vertical* TID-lists (T_{at}).
3. Esta etapa utiliza a mineração de regras de associação interníveis, ou Cross-level association rule mining (CAR) em inglês, que gera regras de associação contendo o targetAttr na conseqüência das regras usando o algoritmo FARAMS (LEUNG, CHAN E CHUNG, 2006).
4. A conseqüência das regras geradas no passo anterior são substituídas pelo *targetItem.*
5. As regras são armazenadas em uma base de regras.

Gerar recomendações

Para gerar as recomendações o algoritmo segue os passos seguintes:

1. Extrair da base de regras somente as que são relevantes para o usuário ativo. A regra "$p_i \rightarrow p_j$" é considerada relevante se o usuário u já indicou gostar do item p_i anteriormente, mas ainda não avaliou o item p_j. O item p_j é considerado um item recomendável para o usuário ativo.
2. Para cada item recomendável é associado um valor de preferência determinado por uma pontuação de interesse como suporte,

confiança ou valores de correlação, que em geral são indicadores da qualidade das regras. Se o item recomendável p_j aparece na conseqüência de mais de uma regra relevante, serão somadas as pontuações de interesse destas regras para fazer uma predição da preferência do usuário para o item p_j. Quanto maior o valor de preferência obtido por um item recomendável, mais provável que o usuário ativo irá gostar deste item.

Por fim, os N itens com os maiores valores de preferência serão recomendados.

Item	TID-list	Suporte total
m1	u1	1
m2	u2, u3	2
m3	u2, u3	2

(a) Suporte de m1, m2 e m3

Item	TID-list	Suporte total
m3	u2, u3	2
c1	u1, u2, u3	3

(b) Suporte de c1 e m3

Item	TID-list	Suporte total
{m3, c1}	u2, u3	2

(c) Conjunto de itens freqüentes (*minSupp* = 1)

Regra de Associação	Suporte	Confiança
m3 → c1	2/3 = 66.7%	2/2 = 100%

(d) Regra de associação

Figura 14 *Vertical TID-list*
Fonte: Adaptado de Leung, Chan e Chung (2008)

Utilizando o modelo mostrado na Figura 13 e considerando que o item m_1 (*Cold-Start Item*) seja o *targetItem*, que o valor de suporte mínimo

(*minSupp*) seja igual a 1 e o atributo elenco seja usado para gerar recomendação, o algoritmo CLARE irá gerar recomendações da seguinte forma: de acordo com a Figura 13 tanto o item $m1$ quanto $m2$ possuem o atributo elenco "Tom Hanks", denotado como $c1$ na Figura 14. É possível obter as transações mostradas na Figura 14 (b). O suporte (número de transações no qual o um conjunto de itens ocorre como um subconjunto) das transações de $c1$ é a união do suporte das transações de $m1$ e $m2$. Como mostrado na Figura 14 (c) $\{m3|c1\}$ satisfaz o *minSupp*, e baseado nisso a regra de associação "$m3 \rightarrow c1$" com suporte 66.7% e confiança 100% pode ser minerada. Desta forma, se o usuário já indicou preferência pelo item $m3$, pode ser recomendado a ele o item $m1$ dada a regra de associação $m3 \rightarrow c1$.

5.2 Análise dos Algoritmos de FI

Conforme visto na seção 5.1 e exemplificado no Quadro 1 os algoritmos de FI levantados possuem características diferentes, tais como o método de FI utilizado e as técnicas que ajudam na recomendação aos usuários. Esta seção busca fazer a análise destes algoritmos sobre como podem ser utilizados para solucionar o problema *Cold-Start Item*.

Quadro 1 Características das FI levantadas.

Algoritmo	Método de Filtragem	Técnicas
Kim e Li (2004)	FC baseada em modelo (baseada no item)	Algoritmos de clusterização *K-means* e *k-Medoids;* Coeficiente de correlação de Pearson.
Hofmann (2004)	FC baseada em modelo (baseada no item)	Singular Value Decomposition (SVD)
Tiraweerakhajohn e Pinngern (2004)	FH	Mineração de regras de Associação (algoritmo Apriori); Método do cosseno.

Quadro 1, conclusão

Han e Karypis (2005)	FC baseada em modelo (baseada no item)	* Algoritmo de recomendação Top-N baseado em item; ** Regras de associação; ** *Matching rules.*
Christakou e Stafylopatis (2005)	FH (mista)	Redes Neurais Artificiais; Coeficiente de correlação de Pearson.
Liu et al. (2007)	FC baseada em modelo (baseada no item)	TF-IDF; Coeficiente de correlação de Pearson.
Wang e Kong (2007)	FCH	Ontologia; Algoritmo de clusterização K-*means.*
Leung, Chan e Chung (2008)	FC baseada em modelo (baseada no item)	Cross-Level Association RulEs (CLARE); Mineração de regra de associação *fuzzy;* Regras de associação; *Vertical TID-list.*

* Técnicas utilizada somente para o primeiro algoritmo proposto pelos autores.
** Técnicas utilizadas somente para o segundo algoritmo proposto pelos autores.

Dentre todos os algoritmos somente o proposto pelos autores Christakou e Stafylopatis (2005) utiliza o método de FH mista, combinando as recomendações resultantes da parte de FBC com a FC, para enfim fazer uma recomendação final. A técnica de RNA utilizada pode ajudar a recomendar um filme com nenhuma avaliação (*Cold-Start Item*) da seguinte forma: este filme é fornecido como entrada para as redes neurais dos usuários, cuja saída ira indicar se este item é recomendável ou não ao usuário. Porém, o sucesso na recomendação de novos itens dependerá de quantos filmes o usuário avaliou em duas ocasiões: (i) quando o usuário avaliou poucos filmes, serão fornecidos poucos exemplos para treinamento da rede e (ii) quando o usuário avaliou uma quantidade muito grande de filmes, é possível que muitos apresentem as mesmas características, porém sejam diferentes uns dos outros, e assim são avaliados

com valores diferentes pelo usuário. Estas duas situações vão afetar o aprendizado da RNA, que poderá fornecer saídas erradas e resultar na não recomendação do novo filme. Na parte de FC quando um usuário possui poucas avaliações, esses dados não serão suficientes para relacioná-lo com outros usuários (*Cold-Start User*). Logo, embora a combinação da FBC com a FC seja utilizada para que uma supere as limitações da outra, a qualidade e precisão final das recomendações podem ser afetadas quando ambas as partes apresentarem os problemas citados acima.

A utilização das características dos itens em todas as etapas de implementação dos algoritmos propostos por Han e Karypis (2005), fazem com que estes consigam solucionar o problema *Cold-Start Item*. Pois, no primeiro algoritmo, dadas as características dos itens que o usuário ativo já comprou, é possível encontrar todos os itens já vendidos em um determinado período com as mesmas características que estes. Estes itens por sua vez podem ser utilizados para encontrar itens candidatos a serem recomendados, utilizando para isso o modelo que expressa as relações entre todos os itens construído através do algoritmo de recomendação *Top-N* baseado em item (Karypis, 2001). Por fim, os itens disponíveis atualmente, com características similares aos itens candidatos a serem recomendados (incluindo novos itens) serão recomendados. Até mesmo um item considerado novo por possuir poucos dados de compra, se estiver presente no cesto de compra do usuário ativo será utilizado no processo de recomendação. O segundo algoritmo proposto também soluciona o problema *Cold-Start Item*, sem a necessidade de se criar um modelo que expresse as relações entre os itens, mas somente utilizando regras de associação.

Ambos os algoritmos de Tiraweerakhajohn e Pinngern (2004) e CLARE utilizam a técnica de mineração de regras de associação. O resultado final do método de Tiraweerakhajohn e Pinngern (2004) é a predição de avaliação do

usuário para itens similares ao que ele já avaliou que foram encontrados utilizando a mineração de regras de associação. Logo, um novo item só poderá ser recomendado, se o algoritmo encontrar similaridades entre ele com os já avaliados pelo usuário. Caso contrário, a contribuição do algoritmo será para se criar uma nova matriz completa, que compreende as avaliações reais dos usuários, com as avaliações resultantes do cálculo das predições. Isto indica que o algoritmo é mais adequado para solucionar o problema da esparsidade, descrito na seção 2.2.2.2, e não o *Cold-Start*.

Já no algoritmo CLARE seu objetivo principal é solucionar o problema *Cold-Start Item*, e ele se mostra mais eficiente. Quando um novo item não pode ser relacionado a nenhum outro, CLARE infere preferências a ele através dos valores de seus atributos, que ele compartilha com os outros itens. Utilizando a base de dados MovieLens para os experimentos, que contém 1682 filmes, 943 usuários com um total de mais de 100.000 avaliações, os resultados mostraram que a escolha dos atributos dos itens produz resultados diferentes. O atributo Gênero produziu a pior qualidade comparada aos outros atributos, devido a sua alta generalização (na base de dados MovieLens Gênero possuía 18 valores diferentes). A melhor qualidade é alcançada através do atributo Diretor, o atributo mais específico (na base de dados MovieLens 70% dos diretores dirigiram somente um filme). Porém, esta característica fez com que o atributo Diretor produzisse a pior cobertura (percentagem de itens no qual as recomendações podem ser fornecidas). Para solucionar este problema o atributo Diretor foi combinado com o atributo Elenco que produziu a segunda melhor qualidade na recomendação de itens que sofrem do problema *Cold-Start Item*. A Figura 15 mostra qual a precisão e cobertura das recomendações de acordo com as características dos filmes.

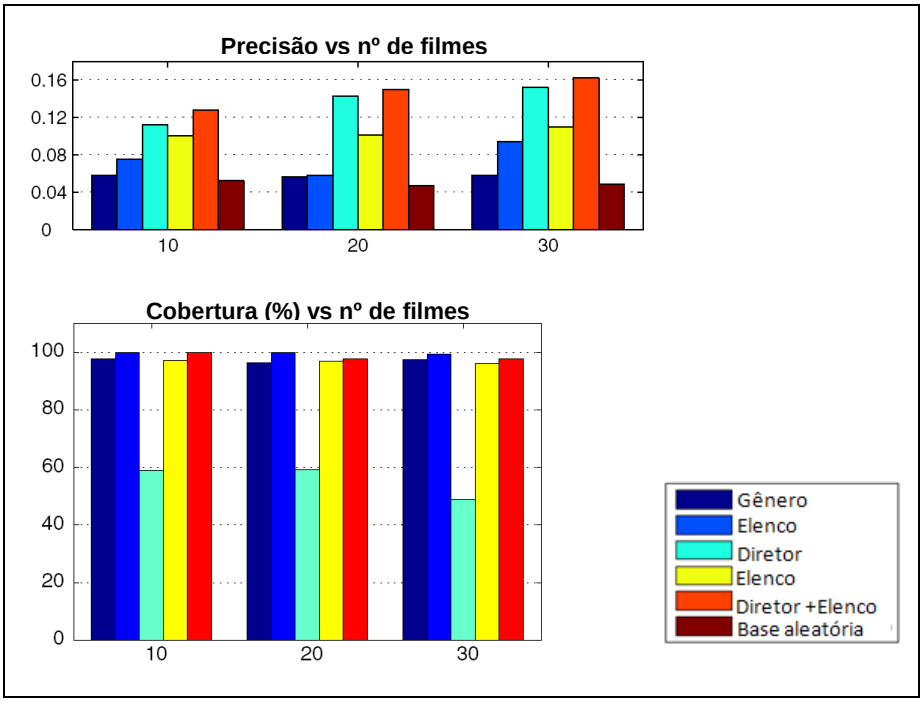

Figura 15 Precisão e cobertura do algoritmo CLARE
Fonte: Adaptado de Leung, Chan e Chung (2008)

Através de seus algoritmos de clusterização, o método de Kim e Li (2004) podem solucionar o problema *Cold-Start Item* da maneira seguinte. Através de seus atributos e utilizando o algoritmo de clusterização *k-means* um novo item poderá ser associado a grupos de itens. Feito isso será criada uma nova matriz usuário-item, compreendendo tanto as avaliações dos usuários, como os grupos de itens. Assim, o novo item que antes não poderia ser associado aos outros itens por não possuir nenhuma avaliação, agora pode ser associadao de acordo com seus atributos. Através desta nova matriz e utilizando o algoritmo *k-medoids*, o novo item será associado ao *cluster* onde exista itens bastante similares a ele. Por fim, é possivel predizer o voto de um usuário para este novo item, logo ele poderá ser recomendado para algum usuário. Utilizando

como experimento filmes extraídos da base de dados MovieLens[9] e a métrica Mean Absolute Error (MAE) para medir a qualidade das predições, Kim e Li (2004) mostraram que a qualidade das predições é alcançada quando o número de comunidades de itens criada através do algoritmo de clusterização *k-medoids* é igual a 70, conforme Figura 16. Nesta figura quanto menor o valor da métrica MAE maior é a precisão do algoritmo.

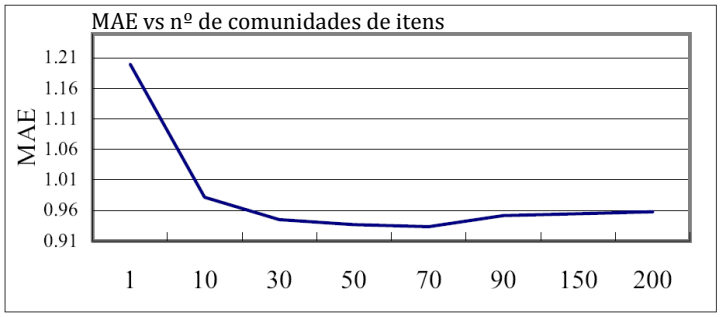

Figura 16 Número de comunidades de itens
Fonte: Kim e Li (2004)

Além disso, a combinação entre os atributos dos filmes (gênero, atores, diretor) forneceu um melhor desempenho para o algoritmo conforme mostra a Figura 17.

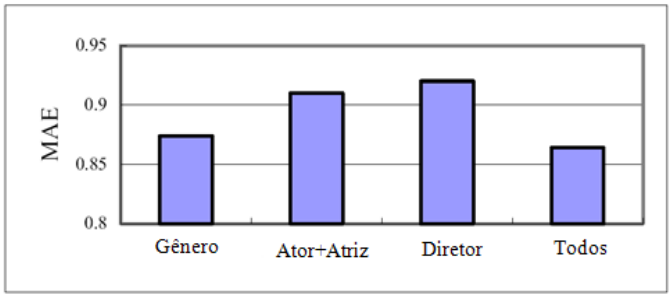

Figura 17 Comparação entre os atributos dos itens
Fonte: Adaptado de Kim e Li (2004)

[9] http://movielens.umn.edu

O algoritmo de Hofmann (2004) é semelhante ao algoritmo de Kim e Li (2004) por também utilizar uma varíavel escondida z. Porém, para Kim e Li (2004) z representa uma comunidade de itens e para Hofmann (2004) clusters de usuários. Este algoritmo pode ser utilizado diante do problema *Cold-Start User* uma vez que um novo usuário pode ser associado a algum *cluster* e assim é possível estimar sua avaliação para itens com base nos itens já avaliados pelos usuários membros do mesmo *cluster*. Além disso, este algoritmo está relacionado com a técnica algébrica de fatoração de matrizes denominada *Singular Value Decomposition* (SVD) que é capaz de descobrir através da matriz usuário-item preferências dos usuários por aspectos dos itens, como exemplo características dos filmes. Assim uma generalização deste algoritmo pode ser utilizada para conseguir criar um modelo que mostre a correlação entre os itens, e assim fazer com que ele seja usado para recomendação de novos itens (*Cold-Start Item*).

O objetivo do algoritmo de FI propostos por Liu, Wang, Fang, Mi (2007) não é solucionar o problema *Cold-Start*, mas sim o problema da esparsidade. Porém, nada impede que partes de seu algoritmo sejam combinadas com outros algoritmos para assim solucionar o problema *Cold-Start Item*. Como exemplo, para um novo item que não possui nenhum dado de compra ou avaliação, utilizando um documento com suas características este poderá ser relacionado com outros itens utilizando-se a técnica TF-IDF criando-se uma matriz de similaridade item-item conforme propõe este algoritmo. Diferentes técnicas podem ser utilizadas sobre esta matriz de similaridade para enfim fazer predições ou lista de recomendações a determinado usuário.

O algoritmo de Wang e Kong (2007) pode ser utilizado para solucionar o problema *Cold-Start User*, pois mesmo para um usuário sem nenhuma avaliação ou dados de compra podem-se encontrar usuários similares a ele

através de suas informações demográficas. A construção de uma ontologia de domínio faz deste algoritmo muito complexo, visto que a construção de uma ontologia de domínio, como exemplo de um SR de filmes, pode requerer muito tempo, pois a quantidade de filmes e categorias na qual estes itens pertencem é muito grande.

5.3 Resultados

Através da análise das características dos algoritmos de FI levantados neste trabalho, e expostas na seção 5.1 foi possível identificar quais podem ser utilizados para solucionar o problema *Cold-Start Item* enfrentado pela empresa Chaordic Systems.

Dentre todos os algoritmos os mais promissores são os propostos por Kim e Li (2004), Han e Karypis (2005) e Leung, Chan e Chung (2008). Estes algoritmos se mostraram efetivos na solução do problema *Cold-Start Item*, e todos utilizando avaliações ou dados de compra dos usuários conseguem criar um modelo que expressa as similaridades entre os itens através de suas características. A partir deste modelo, estes algoritmos conseguem recomendar novos itens com base em suas características.

Quadro 2 Comparação entre os algoritmos mais promissores.

Algoritmo	**Dados de entrada**	**Saída do algoritmo**
Kim e Li (2004)	Avaliações	Predição
Han e Karypis (2005)	Compra	Lista de recomendação
Leung, Chan e Chung (2008)	Avaliações	Lista de recomendação

Além das diferentes técnicas utilizadas no processo de recomendação estes algoritmos também se diferem quanto aos dados de entrada e saídas conforme Quadro 2. Logo para SR que utilizam dados implícitos os algoritmos

de Han e Karypis (2005) são os mais indicados, caso contrário devem ser utilizados os algoritmos de Kim e Li (2004) e Leung, Chan e Chung (2008).

6 CONCLUSÕES

A técnica de FI é fundamental para os SR, pois através dela é possível descobrir relações como a compra de um item que sempre leva à compra de outro item, ou usuários com histórico de compras comuns. Estas descobertas da FI sobre os dados permitem a predição ou recomendação de itens que sejam relevantes aos usuários. Porém, a qualidade das recomendações é afetada quando um item é novo no sistema e conseqüentemente não possui nenhum ou poucos dados de compra ou avaliação. Este problema, denominado *Cold-Start Item*, faz com que não seja possível relacionar o novo item com os demais resultando na sua não recomendação.

Neste trabalho foi realizado um levantamento do Estado da Arte de algoritmos de FI que com o auxílio de diferentes técnicas como clusterização, regras de associação, redes neurais artificiais e ontologia podem solucionar o problema *Cold-Start Item*. Foi apresentada uma análise destes algoritmos identificando seus prós e contras e se são totalmente aplicáveis na solução do problema *Cold-Start Item*. Como resultado desta análise os algoritmos propostos por Kim e Li (2004), Han e Karypis (2005) e Leung, Chan e Chung (2008) foram selecionados como os mais promissores e efetivos na solução do problema *Cold-Start Item*. No algoritmo de Kim e Li (2004) os itens são classificados em grupos e as predições são feitas para os usuários considerando a distribuição gaussiana de suas avaliações. Os algoritmos de Han e Karypis (2005) por sua vez utilizam tanto o algoritmo de recomendação Top-N baseado em item como regras de associação para recomendar novos itens. O algoritmo Cross-Level Association RulEs (CLARE) proposto por Leung, Chan e Chung (2008) opera sobre um modelo que compreende as relações existentes entre

usuário-item e item-item, e a recomendação de novos itens é realizada através de suas características.

Apesar de nenhum dos algoritmos considerados mais promissores terem sido implementados e testados no contexto deste trabalho, acredita-se que os mesmos podem ser utilizados pela empresa Chaordic Systems, cujos SR desenvolvidos sofrem com o problema *Cold-Start Item*. Como trabalhos futuros têm-se a seleção final, implementação e testes dos algoritmos propostos neste trabalho. Vale, neste momento, ressaltar que a solução a ser implementada deverá considerar as nuances dos dados utilizados pela empresa Chaordic Systems bem como a realidade mercadológica da mesma, unindo características dos algoritmos estudados e daqueles já implementados pela empresa, de forma a compor uma solução eficiente.

7 REFERÊNCIAS

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. In: IEEE Transactions on Knowledge and Data Engineering, v. 17, n. 6, p. 734-749, jun. 2005.

AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. Mining Association Rules between sets of items in large databases. IN: INTERNATIONAL CONFERENCE MANAGEMENT OF DATA (SIGMOD-93), p.207-216, 1993.

AGRAWAL, R.; SRIKANT, R. Fast Algorithms for mining association rules. In: PROCEEDINGS OF THE 20TH INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES, n. 20, Santiago, Chile, p.487-499,1994.

ALBADVI, A.; SHAHBAZI, M. A hybrid recommendation technique based on product category attributes. *Expert Systems with Applications: An International Journal*, v. 36, n. 9, p. 11480-11488, nov. 2009.

BERNARTT, J. L. V. **Um sistema de recomendação baseado em filtragem colaborativa.** 2008. 87 p. Dissertação (Mestrado em Engenharia Elétrica) - Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, 2008.

BREESE, J.; HECKERMAN, D.; KADIE, C. Empirical analysis of predictive algorithms for collaborative Filtering. In: PROCEEDINGS OF THE 14TH CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, ACM, n. 14, , New York, NY, USA, p. 43-52, 1998.

BURKE, R. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, v. 12, n. 4, p. 331-370, nov. 2002.

CHRISTAKOU, C.; STAFYLOPATIS, A. A Hybrid Movie Recommender System Based on Neural Networks. In: PROCEEDINGS OF THE 5TH INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS, Wroclaw, Poland, p. 500-505, 2005.

DESHPANDE, M.; KARYPIS, G. Item-based top-N recommendation algorithms. *ACM Transactions on Information System*, v. 22, p. 143-177, jan. 2004.

GAZZANAL, P. P.; SILVEIRA, S. R. RecomenTur - Sistema de Recomendação para a Área de Turismo. In: ANAIS DO VIII SEMINÁRIO DE INFORMÁTICA - RS, Torres, RS, 2009.

GOLDBERG, D. et al. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, v. 35, n. 12, p. 61-70, dec. 1992.

HAN, E-H. S.; Karypis, G. Feature-based recommendation system. In: CIKM '05: PROCEEDINGS OF THE 14TH ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, New York, NY, USA, p 446-452, 2005.

HAN, J.; KAMBER, M. Data mining: Concepts and Techniques. 2.ed. New York: Morgan-Kaufman, 2000. 800p.

HANANI, U.; SHAPIRA, B.; SHOVAL, P. Information filtering: Overview of issues, research and systems. *User Modeling and User Adapted Interaction*, v. 11 n. 3, p. 203-259, aug. 2001.

HOFMANN, T. Latent Semantic Models for Collaborative Filtering. *ACM Transactions on Information Systems*, v. 22, n. 1, p. 89-115, jan. 2004.

KAJIMOTO, A. P. K. et al. **Sistemas de recomendação de notícias na Internet baseados em filtragem colaborativa.** 2008. 37p. Trabalho de Formatura Supervisionado - Instituto de Matemática e Estatística, Universidade de São Paulo, 2007.

KARYPIS, G. Evaluation of item-based top-*n* recommendation algorithms. In: PROCEEDINGS OF THE ACM CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, Atlanta, Georgia, USA, 2001.

KIM, B. M.; LI, Q. Probabilistic Model Estimation for Collaborative Filtering Based on Items Attributes. In: PROCEEDINGS OF THE 2004 IEEE/WIC/ACM INTERNATIONAL CONFERENCE ON WEB INTELLIGENCE, Beijing, China, p.185-191, 2004.

KIM, B. M., et al. A new approach for combining content-based and collaborative filters. *Journal of Intelligent Information Systems*, v. 27, n. 1, p. 79-91, jul. 2006.

LEUNG, C. W. K.; CHAN, S. C. F.; CHUNG F. L. A collaborative filtering framework based on fuzzy association rules and multiple-level similarity. *Knowledge and Information Systems*, v. 10, n. 3, p. 357-381, 2006.

LEUNG, C. W. k.; CHAN, S. C-f.; CHUNG, F. L. An empirical study of a cross-level association rule mining approach to cold-start recommendations. *Knowledge-Based Systems*, v. 21, p. 515-529, oct. 2008.

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

LIU, J. et al. An optimized collaborative filtering approach combining with item-based prediction. In: PROCEEDINGS OF THE 11TH INTERNATIONAL CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK IN DESIGN, Melbourne, Australia, p. 157-161, 2007.

LOPES, G. R.; SOUTO, M. A. M; OLIVEIRA, J. P. M. Sistema de recomendação para Bibliotecas Digitais sob a perspectiva da Web Semântica. In: II WORKSHOP DE BIBLIOTECAS DIGITAIS, WDL; SBBD/SBES, Florianópolis, SC, p. 21-30, 2006.

MIDDLETON, S. E.; ALANI, H.; ROURE D. C. (2002). Exploiting Synergy Between Ontologies and Recommender Systems. In: SEMANTIC WEB WORKSHOP, Hawaii, USA, 2002.

MOONEY, R. J.; BENNETT, P. N.; ROY, L. Book recommending using text categorization with extracted information. In: AAAI-98/ICML-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION AND THE AAAI-98 WORKSHOP ON RECOMMENDER SYSTEMS, New York, NY, USA, p. 1-7, 1998.

PARK, S. T.; CHU, W.; Pairwise preference regression for cold-start recommendation. In: RECSYS '09: PROCEEDINGS OF THE THIRD ACM CONFERENCE ON RECOMMENDER SYSTEMS, New York, p. 21-28, 2009.

PAZZANI, M. J. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, v. 13, n.5, p. 393-408, dec. 1999.

PENNOCK et al. Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-Based Approach. In: PROCEEDINGS OF THE 16TH CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, Estocolmo, p. 473-480, 2000.

REATEGUI, E. B.; CAZELLA, S. C. Sistemas de Recomendação. In: XXV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2005, São Leopoldo, p. 30-348, 2005.

SALTON, G. Automatic text processing, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1989.

SARWAR, B. M. et al. Analysis of Recommendation Algorithms for ecommerce. In: ACM CONFERENCE ON E-COMMERCE, n. 2, 2000, Minneapolis, p. 158-167, 2000.

SARWAR, B. et al. Item-Based Collaborative Filtering Recommendation Algorithms. In: PROCEEDINGS OF THE 10TH INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, ACM Press, n. 10, 2001, Hong Kong, p. 285-295, 2001.

SCHAFER, J.; KONSTAN, J.; RIEDL, J. Recommender systems in e-commerce. In: PROCEEDINGS OF THE 1ST ACM CONFERENCE IN ELECTRONIC COMMERCE, Denver, Colorado, USA. p. 158-166, 1999.

SHARDANAND, U.; MAES, P. Social information filtering: Algorithms for automating "word of mouth". In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, CHI, 1995, Denver, US, p. 210-217, 1995.

TIRAWEERAKHAJOHN, C.; PINNGERN, O. Finding Item Neighbors in Item-based Collaborative Filtering by Adding Item Content. In: PROCEEDINGS OF THE 8TH INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION, Kunming, China, p. 1674-1678, 2004.

WANG, R-Q.; KONG, F-S. Semantic-enhanced personalized recommender system. In: PROCEEDINGS OF THE 6TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS (ICMLC-07), Hong Kong, China, 2007, p. 4069–4074.

XUE et al. Scalable collaborative filtering using cluster-based smoothing. In: PROCEEDINGS OF THE 28TH ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, Salvador, Brasil, p. 114-121, 2005.

ZAKI, M. J. Scalable algorithms for association mining. In: IEEE Transactions on Knowledge and Data Engineering, v. 12, p.372–390, 2000.