ANÁLISE DE TÉCNICAS DE FILTRAGEM COLABORATIVA EM UM SISTEMA DE RECOMENDAÇÃO INTER-APLICAÇÕES

Trabalho de Conclusão de Curso

Engenharia da Computação

Arthur Inácio do Nascimento
Orientador: Prof. Dr. Byron Leite Dantas Bezerra

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação

ARTHUR INÁCIO DO NASCIMENTO

ANÁLISE DE TÉCNICAS DE FILTRAGEM COLABORATIVA EM UM SISTEMA DE RECOMENDAÇÃO INTER-APLICAÇÕES

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Junho de 2011.

De acordo

Recife

____/_____/_____

_____
Orientador da Monografia

Aos meus pais, irmã e todos aqueles que torcem por mim.

Agradecimentos

Agradeço com muita satisfação aos meus pais e minha irmã, pelo apoio em todos os momentos importantes. Agredeço pelo incentivo dado por eles, desde dos tempos de colégio até o período da minha formação acadêmica.

Agradeço aos professores da graduação por terem participado da minha formação como profissional da área de Engenharia da Computação, repassando seus conhecimentos, experiências e ensinamentos. Agradeço ao meu orientador professor Byron Leite, pela sua preocução na escolha do tema, disponibilidade e auxílio na construção deste trabalho.

Agradeço aos meus colegas de turma, que vivenciaram ao meu lado vários momentos dentro da faculdade e caminharam junto nesta difícil jornada.

Por fim, agradeço acima de tudo por ter chegado ao final deste curso e ter completado mais uma etapa da minha vida.

Resumo

Sistemas de recomendação utilizam técnicas de filtragem de informação para recomendar possíveis itens que o usuário que interage com esse sistema, possa se interessar. Os algoritmos usados nas técnicas de filtragem colaborativa comumente apresentam diferentes desempenhos em um tipo de ambiente em que o sistema de recomendação está envolvido. Esses algoritmos podem ser comparados e analisados em um mesmo ambiente utilizando uma métrica que avalia a qualidade das recomendações geradas pelo sistema de recomendação. Sob essa perspectiva que o trabalho proposto foi construído, para realizar uma análise de técnicas de filtragem colaborativa em vários cenários, incluindo a análise em um ambiente de recomendação inter-aplicações. A abordagem inter-aplicações é basicamente um sistema de recomendação que se utiliza dos históricos de avaliações de itens de um usuário em outros sistemas de recomendação que ele também utiliza, para construir um perfil mais robusto e sugerir itens personalizados. Foi desenvolvido neste trabalho um ambiente que simulou um sistema de recomendações inter-aplicações para realizar os experimentos propostos. A integração entre as aplicações foi dada através de um *Web Service* que utilizou a arquitetura orientada a serviços. A criação dos possíveis cenários encontrados em um sistema de recomendação se deu levando em consideração os seguintes fatores: número de usuários disponíveis, o número de itens avaliados por estes usuários e número de aplicações usadas. As técnicas de filtragem colaborativa foram baseadas na distância euclidiana, no coeficiente de correlação de Pearson e em perfis simbólicos modais. Estas três técnicas foram submetidas a esses cenários para serem confrontadas e analisadas. Com isso foram descobertos algoritmos que alcançam melhor desempenho em um determinado cenário com ou sem a influência de uma abordagem inter-aplicações.

Abstract

Recommendation systems use information filtering techniques to recommend possible items to the user who interacts with this system, might be interested upon. The algorithms used in collaborative filtering techniques commonly have differents performances depending on the type of environment where the recommendation system is involved. These algorithms can be compared and analyzed in a single environment using a metric that evaluates the quality of recommendations generated by the recommendation system. From this perspective the proposed work was built to conduct a collaborative filtering techniques in various scenarios, including the analysis in an environment of inter-recommendation application. The inter-application approach is basically a recommendation system that uses the historical ratings of items from a user in other recommendation systems that he uses to build a more robust profile and suggest personalized items. It was developed in this work environment that simulated a system of inter-application recommendations for achieving the proposed experiments. The integration between applications was given through a web service that used the service-oriented architecture. The creation of scenarios found in a recommendation system was made taking into consideration the following factors: number of available users, the number of items assessed by these users and number of applications used. Collaborative filtering techniques were based on Euclidean distance, the Pearson correlation coefficient and SDA. These three techniques were subjected to these scenarios to be compared and analyzed. Thus were discovered algorithms that achieve better performance in a given scenario, with or without the inflluence of a inter-applications approach.

Sumário

CAPÍTULO 1 INTRODUÇÃO 1

1.1 MOTIVAÇÃO E PROBLEMA 1

1.2 OBJETIVOS E METAS 2

1.3 ESTRUTURA DO DOCUMENTO 3

CAPÍTULO 2 FUNDAMENTAÇÃO TEÓRICA 4

2.1 SISTEMAS DE RECOMENDAÇÃO 4

2.1.1 Engenho de Recomendação 5

2.1.2 Filtragem Colaborativa 6

2.1.3 Filtragem Híbrida 8

2.2 ALGORITMOS DE FILTRAGEM COLABORATIVA 9

2.2.1 Coeficiente de Correlação de Pearson 10

2.2.2 Distância Euclidiana 10

2.2.3 Perfis Simbólicos 11

2.2.4 Função de Utilidade 14

2.3 QUALIDADE DAS RECOMENDAÇÕES 15

2.3.1 Métrica de Breese 16

2.4 ABORDAGEM INTER-APLICAÇÕES 17

2.4.1 Conceito Geral 17

2.4.2 Características 18

2.4.3 Arquitetura Orientada a Serviços (SOA) e Web Services 20

CAPÍTULO 3 AMBIENTE EXPERIMENTAL 22

3.1 SISTEMA DESENVOLVIDO 22

3.1.1 Visão Geral 22

3.1.2 Tecnologias 25

3.1.3 Arquitetura 26

3.2 BASE DE DADOS 29

CAPÍTULO 4 ANÁLISE DAS RECOMENDAÇÕES 33

4.1 ESTRUTURA DA ANÁLISE 33
4.1.1 Configuração dos Cenários 33
4.1.2 Procedimento de Execução 35
4.2 EXPERIMENTOS E RESULTADOS 36
4.2.1 Análise de técnicas de filtragem colaborativa utilizando abordagem tradicional 36
4.2.2 Análise de técnicas de filtragem colaborativa utilizando abordagem inter-aplicações 40
CAPÍTULO 5 CONCLUSÕES E TRABALHOS FUTUROS 43
5.1 CONTRIBUIÇÕES 43
5.2 DIFICULDADES E DESAFIOS 44
5.3 TRABALHOS FUTUROS 45
BIBLIOGRAFIA 46

Índice de Figuras

Figura 1. Interação de usuários com um sistema de recomendação. 6

Figura 2. Obtenção dos vizinhos mais próximos do usuário u. 8

Figura 3. Representação gráfica da distância Euclidiana. 11

Figura 4. Exemplo de um ambiente inter-aplicações. 19

Figura 5. Comunicação entre o Web Service e o cliente da aplicação. 21

Figura 6. Menu do ambiente experimental desenvolvido. 23

Figura 7. Tela de recomendações**.** 24

Figura 8. Tela de análise das recomendações. 25

Figura 9. Arquitetura adotada no desenvolvimento do ambiente experimental. 26

Figura 10. Operações presentes no contrato do serviço de integração inter-aplicações. 27

Figura 11. Interfaces usadas dentro do serviço de integração inter-aplicações. 28

Figura 12. Diagrama do banco de dados usado nos experimentos. 30

Figura 13. Resultados da métrica Breese utilizando o algoritmo de correlação de Pearson para o número de vizinhos igual a 5 e 10. 38

Figura 14. Pontos em que os resultados da abordagem inter-aplicações foram visivelmente melhor. 41

Índice de Tabelas

Tabela 1. Matriz de avaliações dos usuários. 12

Tabela 2. Descrições simbólicas dos filmes da Tabela 1. 13

Tabela 3. Perfil simbólico modal de Mateus. 14

Tabela 4. Os cenários envolvidos nas análises. 34

Tabela 5. Nomenclatura usada nos experimentos de análise. 35

Tabela 6. Resultados referentes aos valores da métrica de Breese para um sistema de recomendação utilizando a abordagem tradicional. 37

Tabela 7. Resultados referentes aos valores da métrica de Breese para um sistema de recomendação utilizando a abordagem inter-aplicações. 40

Tabela de Símbolos e Siglas

(Dispostos em ordem alfabética)

- MVC – *Model-View-Controller*
- kNN – *k Nearest Neighborhood*
- SOA – *Service-Oriented Architecture*
- SOAP – *Simple Object Access Protocol*
- WCF – *Windows Communication Foudation*
- WSDL – *Web Service Definition Language*
- XML – *Extensible Markup Language*

Capítulo 1

Introdução

Neste capítulo estão levantados os problemas junto com a motivação para a realização deste trabalho. Estão descritos também os objetivos e metas que foram traçados para a construção do trabalho de conclusão de curso. E por fim, está apresentada a estrutura do documento com os resumos dos capítulos aqui presentes.

1.1 Motivação e Problema

Sistemas de recomendação (SR) podem gerar listas de itens que auxiliam usuários a escolher um item interessante em um determinado contexto[1]. A partir da interação do usuário com este tipo de sistema, itens como filmes, livros ou músicas podem ser recomendados. Uma técnica conhecida e bastante utilizada é a filtragem colaborativa (FC) [1, 2]. A idéia principal deste tipo de técnica é estimar a relevância entre o usuário e um item através da análise das suas experiências [1], juntamente com as avaliações de outros usuários. Este tipo de técnica é amplamente utilizado em sistemas web, como Amazon.com e CDNow.com [2].

Apesar de realizar recomendações interessantes em vários cenários, sistemas web que utilizam a FC possuem alguns problemas inerentes, tais como o problema do novo usuário e do novo item, o problema da esparsidade e o das ovelhas negras [3]. Mediante a estes problemas, pode-se perceber que o nível de qualidade das recomendações depende muito da quantidade de informações acerca do usuário.

Sistemas que utilizam uma abordagem inter-aplicações tentam diminuir estes problemas utilizando os perfis de usuários em outras aplicações para realizar recomendações dentro da aplicação que utilizou o serviço web de integração [4]. Os sistemas que fazem uso desta abordagem possuem a vantagem de, através deste serviço, utilizar as experiências dos usuários em outras aplicações para auxiliar suas recomendações. Outra forma de lidar com esses problemas é utilizar métodos de

filtragem híbrida, que combinam geralmente duas técnicas de filtragem para gerar uma saída simples para um sistema de recomendação. Este tipo de técnica tenta atenuar limitações apresentadas pelas estratégias de filtragem como FC e filtragem baseada em conteúdo [5].

Visto os problemas citados e as abordagens que tentam reduzir estas limitações dos sistemas que utilizam FC, vários cenários podem ser observados nos SR. Cenários estes em que o número de usuários, o número de itens avaliados por estes usuários, número de aplicações utilizadas e os tipos de técnicas implementadas, são fatores que podem influenciar a qualidade de uma lista de recomendações gerada pelos SR.

1.2 Objetivos e Metas

O foco deste trabalho é implementar três técnicas de filtragem colaborativa afim de se comparar o comportamento destes algoritmos e avaliar a qualidade de suas recomendações dentro de diversos cenários. Estes cenários levarão em consideração fatores como: número de usuários disponíveis para a geração da lista de recomendação, quantidade de itens que foram avaliados por estes usuários e número de fontes utilizadas pelo sistema. Este último fator diz respeito às recomendações que serão feitas utilizando uma única aplicação ou utilizando uma abordagem inter-aplicações. Os seguintes objetivos específicos foram traçados:

- Realizar estudos sobre sistemas de recomendação e técnicas de filtragem colaborativa, para auxiliar na implementação e construção do ambiente experimental.
- Implementar os algoritmos de filtragem colaborativa, que serão baseados na distância euclidiana, no coeficiente de correlação de Pearson e em perfis simbólicos.
- Projetar e desenvolver um ambiente computacional para a análise dos algoritmos de filtragem colaborativa. O ambiente desenvolvido usará um *Web Service* utilizando conceitos de SOA[6] e suportará recomendações utilizando uma ou mais aplicações.

- Simular vários cenários de sistemas de recomendação e analisar os resultados gerados pelo ambiente computacional desenvolvido para a medição de qualidade das recomendações.

1.3 Estrutura do Documento

Este documento foi dividido em cinco capítulos, resumidos a seguir:

- **Capítulo 1: Introdução**

Contém o texto introdutório do trabalho. Neste capítulo são apresentados os problemas, as motivações e os objetivos gerais e específicos do estudo realizado;

- **Capítulo 2: Fundamentação Teórica**

Reúne os principais conceitos necessários para a fundamentação teórica e compreensão do trabalho proposto. Para tal, são explicados os conceitos de sistema de recomendação, os algoritmos de filtragem colaborativa utilizados, como se avalia a qualidade de recomendações e sistemas de recomendação que utilizam abordagem inter-aplicações.

- **Capítulo 3: Ambiente Experimental**

Neste capítulo é apresentado o ambiente em que foram realizadas as recomendações e a análise das técnicas de filtragem colaborativa. Contém também detalhes da implementação do serviço de integração, arquitetura do sistema e repositório de conhecimento.

- **Capítulo 4: Análise das Recomendações**

Este capítulo mostra os procedimentos de como os cenários foram criados e configurados. Apresenta também todos os experimentos realizados acerca dos cenários propostos de um sistema de recomendação. Os resultados destes experimentos foram analisados e usados para a extração de conclusões.

- **Capítulo 5: Conclusões e Trabalhos Futuros**

Apresenta a conclusão do trabalho enfatizando as contribuições realizadas e enunciando possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo consiste da apresentação dos conceitos básicos necessários para o entendimento do experimento e análises realizadas. Primeiramente é apresentado, o que é um sistema de recomendação e algumas técnicas de filtragem. Em seguida, faz-se a descrição dos algoritmos de filtragem colaborativa utilizados no experimento. Logo depois, é explicado como se avalia a qualidade das recomendações e a métrica usada neste trabalho. E por último, os conceitos do que vem a ser um sistema de recomendação inter-aplicações e suas características principais.

2.1 Sistemas de Recomendação

Sistemas de recomendação podem ser encontrados em várias aplicações que utilizam grandes coleções de itens, como por exemplo, *e-commerces*. Estes sistemas tentam auxiliar o usuário a encontrar itens que lhe agrade através de uma indicação personalizada representada por uma lista de itens [7]. Esse tipo de aplicação é bastante usada em sistemas de comércio eletrônico para indicação de produtos como: filmes, livros ou cd's [2].

Hoje em dia, o usuário que navega em um sistema web está exposto a uma grande quantidade de informações dos mais diferentes gêneros. Empresas de comércio eletrônico que trabalham com um grande número de produtos, precisam fidelizar o cliente e fazer com que ele esteja sempre comprando seus produtos. Os usuários que utilizam esse tipo de sistema dificilmente conseguem ver todos os itens disponíveis, e na maioria das vezes, utilizam um tempo curto para esse tipo de atividade. Uma estratégia para lidar com esse quadro é utilizar sistemas de recomendação como mecanismos automatizados que tratem as informações disponíveis de uma forma inteligente e personalizada [8].

As principais funcionalidades de sistemas de recomendação são: recomendar uma lista de itens ao usuário e predizer uma nota a um item não avaliado pelo

usuário. Estas duas funcionalidades são conhecidas como, *Find Good Items e Annotation in Context,* respectivamente. A funcionalidade de recomendar uma lista personalizada a um usuário pode ser descrita como uma lista gerada pelo sistema, onde esses itens estão ordenados pela sua utilidade, ou seja, pelo grau de relevância para o usuário. A funcionalidade de predizer a nota diz respeito à capacidade do sistema de recomendação em estimar a relevância de um item para um usuário. *Find Good Items e Annotation in Context* são as duas funcionalidades mais usadas em sistemas de recomendação e estão bastante interligadas [9]. Este trabalho focará nestas duas funcionalidades para a geração e análise das recomendações.

2.1.1 Engenho de Recomendação

Para integrar modelos de recomendação, o sistema deve apresentar três características principais: prover interação do usuário com o sistema, armazenar pontos relevantes e contribuições dos usuários e recomendar conteúdo personalizado a partir dos perfis de cada usuário.

A interação do usuário com esse tipo de sistema pode ser feita através de opiniões e preferências fornecidas pelo usuário acerca dos produtos disponíveis no sistema em questão. Esta avaliação pode ser representada por uma nota, e fornecida utilizando a interface do sistema que deseja recomendar itens. A interface do sistema influencia diretamente a satisfação do cliente, consequentemente a fidelização dele [10].

O engenho de recomendação utiliza as informações adquiridas dos usuários armazenadas no banco de dados para elaborar os perfis, que juntamente com técnicas de computação inteligente (CI) serão capazes recomendar algum item ou gerar uma lista de itens aos usuários. Todos os dados relacionados aos usuários são mantidos na base de conhecimento, e através de uma modelagem realizada pelo sistema que utiliza o engenho de recomendação, eles são manipulados e assim gerarão as recomendações. O conteúdo do modelo de usuários precisa ser organizado de forma que possa ser facilmente obtido e comparado. O nível de estruturação do modelo influenciará na complexidade das transformações e manipulações feitas pelo engenho [10, 11].

Todo esse procedimento tem como finalidade produzir um conteúdo personalizado que seja relevante para o usuário. A **Figura 1** ilustra o processo descrito para que a recomendação seja realizada, onde a linha tracejada representa a obtenção e construção dos perfis dos usuários a partir da base de dados do sistema:

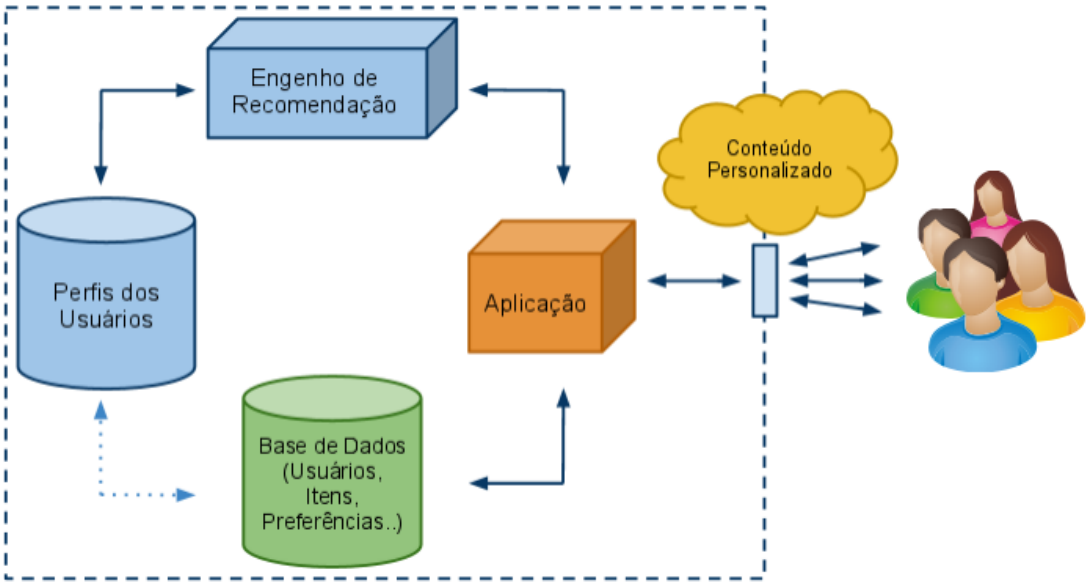

Figura 1. Interação de usuários com um sistema de recomendação.

2.1.2 Filtragem Colaborativa

Uma técnica de filtragem de informação muito usada em sistemas de recomendação é a filtragem colaborativa (FC). A idéia principal deste tipo de técnica é estimar a utilidade de um item a um usuário através da análise de suas experiências juntamente com as avaliações feitas por outros usuários do sistema [1] [2]. Comumente o cenário que envolve a FC é composto por: uma lista de usuários $\boldsymbol{U}$, uma lista de itens $\boldsymbol{I}$ e um conjunto de avaliações acerca dos itens pertencentes a $\boldsymbol{I}$. Essas avaliações são feitas durante as interações dos usuários $\boldsymbol{u} \in \boldsymbol{U}$ com o sistema que utiliza FC [3]. As principais vantagens de técnicas de FC é que elas são completamente independentes de qualquer representação de objetos que serão recomendados, e funciona bem para objetos complexos. Além disso, são métodos que apresentam um grande nível de reusabilidade devido ao padrão do algoritmo desenvolvido [12].

Apesar de a FC possuir essas características que a fazem ser muito utilizada e realizar recomendações interessantes em vários cenários, os sistemas web que utilizam esta técnica possuem alguns problemas inerentes, tais como o problema do novo usuário e do novo item, o problema da esparsidade e o das ovelhas negras.

O problema do novo usuário e o do novo item compartilha da mesma idéia. Existem poucas informações sobre o usuário ou item, consequentemente há uma diminuição da precisão das predições. Um item recém cadastrado no sistema não terá nenhumas ou poucas avaliações dos usuários do sistema. Caso um novo usuário entre no sistema, ele não terá feito avaliações sobre os itens do sistema, dificultando assim a criação do seu perfil para as futuras predições.

A esparsidade na base de conhecimento diz respeito à avaliação feita pelos usuários a vários itens diferentes, existindo assim, itens não avaliados na base de dados ou pouco avaliados.

E o último problema citado, o das ovelhas negras, pode ser encarado como uma classe de usuários que possuem preferências que não seguem nenhum padrão ou não se encaixam em nenhuma classe de preferências de usuários [3]. Esses problemas podem ser atenuados utilizando técnicas de filtragem híbrida ou abordagens inter-aplicações. Estes tópicos serão discutidos mais adiante.

A forma mais comum de aplicação da técnica de FC é através do método baseado nos vizinhos mais próximos, também conhecido como kNN (do inglês, *K Nearest Neighborhood*). O método kNN identifica usuários semelhantes de acordo com o histórico de preferências de cada um deles. Segue abaixo a descrição do algoritmo usado para obter as recomendações:

i. Calcular a similaridade do usuário $\boldsymbol{u}$ com os outros usuários presentes no sistema utilizando um algoritmo de FC
ii. Identificar os k usuários que mais se assemelham com o usuário $\boldsymbol{u}$
iii. Predizer uma nota para um item específico ou gerar uma lista personalizada de itens para o usuário $\boldsymbol{u}$.

No primeiro passo, o cálculo é feito usando um algoritmo de filtragem colaborativa que irá medir a similaridade entre o usuário $\boldsymbol{u}$ e todos os outros usuários. Os algoritmos de filtragem serão descritos na próxima seção. No passo seguinte, o objetivo principal é encontrar os k usuários mais similares ao usuário $\boldsymbol{u}$.

A lista de usuários deve ser ordenada de forma decrescente com base no valor de similaridade encontrado no primeiro passo. Desta forma, a vizinhança do usuário $\boldsymbol{u}$ é formada a partir do valor determinado para k e pronta para ser utilizada nos cálculos seguintes. No passo três, o conjunto de k usuários mais próximos será utilizado para predizer uma nota ou gerar a lista personalizada de itens através de uma função de utilidade [13, 14, 15]. A **Figura 2** ilustra o procedimento acima:

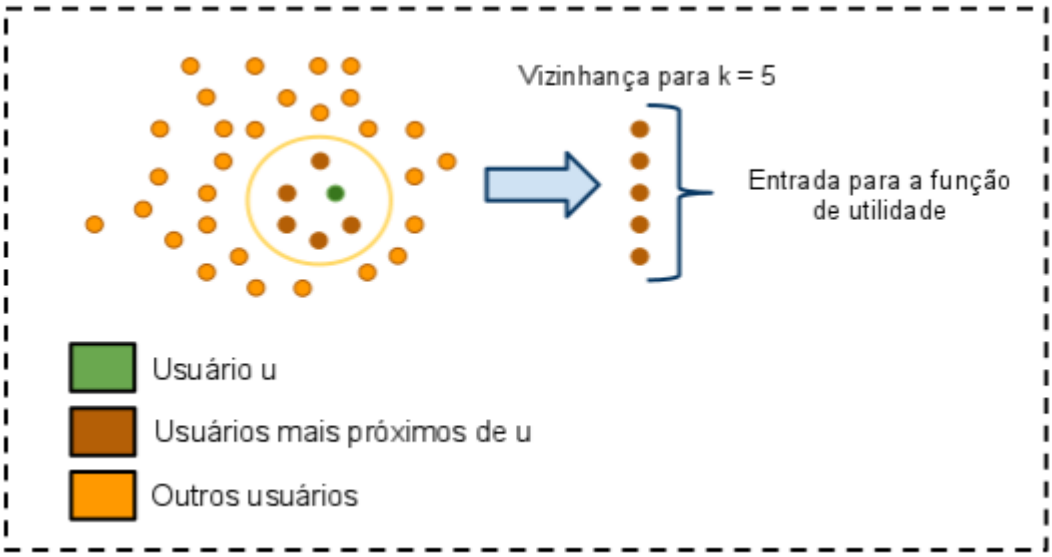

Figura 2. Obtenção dos vizinhos mais próximos do usuário u.

O cálculo da similaridade explicado no procedimento acima pode ser feito de duas maneiras: a *memory-based* ou *model-based*. A abordagem model-based utiliza modelos que a partir dos dados e avaliações dos usuários, realizam as predições. Já a abordagem memory-based usa todo ou uma amostra do banco de dados que possui as relações dos usuários com os itens, para fazer as predições [3]. Este trabalho usará as duas abordagens citadas, a *memory-based* nas técnica de filtragem baseada no coeficiente de correlação de Pearson e na distância euclidiana, e a *model-based* na técnica baseada em perfis simbólicos modais.

2.1.3 Filtragem Híbrida

Sistemas de recomendação que usam filtragem híbrida são sistemas que combinam FC com outra técnica de filtragem, geralmente filtragem baseada em conteúdo, para fazer realizar as recomendações. Sistemas de recomendação que utilizam filtragem baseada em conteúdo analisam o conteúdo da informação textual, tais como documentos, descrições de itens e perfis de usuários, e tentam encontrar

regularidades no conteúdo. O problema clássico desta estratégia de filtragem é a impossibilidade de codificar algumas informações na primeira abordagem. Na tentativa de evitar as limitações das técnicas de filtragem colaborativa e da técnica baseada em conteúdo, um exemplo de filtragem híbrida utilizaria as características positivas desses dois modelos, para assim aumentar o desempenho de suas recomendações [9, 16].

As principais formas de se incorporar as características das técnicas de filtragem colaborativa e baseada em conteúdo em um sistema de recomendação são:

i. Implementar separadamente as duas técnicas e combinar as saídas geradas.
ii. Usar alguns pontos positivos da filtragem colaborativa em uma implementação baseada em conteúdo ou vice-versa.
iii. Construir um modelo que unifique as duas técnicas [17].

O algoritmo de filtragem colaborativa baseado em perfis simbólicos é um exemplo de um algoritmo híbrido que utiliza a terceira maneira de citada acima para o desenvolvimento de suas recomendações. Este algoritmo será detalhado na próxima seção.

2.2 Algoritmos de Filtragem Colaborativa

Como descrito na seção 2.1.2, o método kNN é usado para encontrar os vizinhos mais próximos de um usuário que se pretende predizer a nota de um item ou recomendar uma lista personalizada de itens. Faz-se necessário a utilização de um algoritmo de filtragem para calcular a similaridade entre os usuários e identificar os mais próximos.

Nesta seção serão explicadas os três algoritmos de filtragem que foram implementados, e posteriormente usados na análise de qualidade das recomendações. Todos os algoritmos implementados neste trabalho seguem o padrão de um algoritmo colaborativo baseado no método kNN. A diferenciação acontece na implementação das funções que calculam a similaridade entre os usuários. Estas funções são capazes de gerar uma saída numérica que será usada

para a identificação dos vizinhos mais próximos. As equações descritas a seguir, utilizarão como exemplo, avaliações de itens pertencentes ao conjunto $\boldsymbol{I}$, feitas pelos usuários $\boldsymbol{r}$ e $\boldsymbol{u}$**.**

2.2.1 Coeficiente de Correlação de Pearson

A correlação de Pearson é uma medida de correlação linear entre duas variáveis. Esta medida é utilizada em cálculos estatísticos, além de ser muito usada para calcular similaridade entre usuários em sistemas que utilizam FC. Este coeficiente varia entre -1, indicando uma correlação negativa entre dois usuários, através de 0, indicando ausência de correlação, para 1 indicando uma correlação positiva entre dois usuários [18].

A **Equação 2.1** demonstra como esta medida deve ser calculada. Onde $\vec{\boldsymbol{r}}$ e $\vec{\boldsymbol{u}}$ são, respectivamente, os vetores de avaliações dos usuários $\boldsymbol{r}$ e $\boldsymbol{u}$; $\overline{\vec{\boldsymbol{r}}}$ e $\overline{\vec{\boldsymbol{u}}}$ representam a média aritmética das avaliações realizadas pelos usuários $\boldsymbol{r}$ e $\boldsymbol{u}$, respectivamente; $\vec{\boldsymbol{r}}(\boldsymbol{i})$ é a avaliação do item $\boldsymbol{i}$ realizada pelo usuário $\boldsymbol{r}$; e $\vec{\boldsymbol{u}}(\boldsymbol{i})$ é a avaliação do item $\boldsymbol{i}$ realizada pelo usuário $\boldsymbol{u}$**.**

$$\omega(r,u) = \frac{\sum_{i\in I}\left(\vec{r}(i) - \overline{\vec{r}}\right) * \left(\vec{u}(i) - \overline{\vec{u}}\right)}{\sqrt{\sum_{i\in I}\left(\vec{r}(i) - \overline{\vec{r}}\right)^2 * \sum_{i\in I}\left(\vec{u}(i) - \overline{\vec{u}}\right)^2}} \tag{2.1}$$

2.2.2 Distância Euclidiana

Outra medida usada para calcular a similaridade entre usuários é a distância euclidiana. Esta medida é um caso particular da medida de Minkowski e uma das mais utilizadas para a atividade de agrupamento de dados. A distância euclidiana consisti em calcular a distância direta entre dois pontos a partir da hipotenusa do triângulo formado entre eles [19]. A **Figura 3** ilustra esta operação, onde são formadas linhas ortogonais entre os dois usuário gerando um segmento linear direto entre eles, representado pela hipotenusa, para o cálculo final da similaridade.

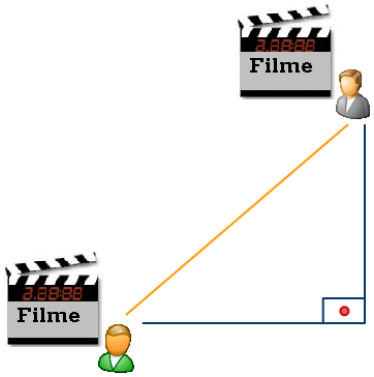

Figura 3. Representação gráfica da distância Euclidiana.

A **Equação 2.2** descreve o cálculo para esta medida, onde $\vec{r}$ e $\vec{u}$ são, respectivamente, os vetores de avaliações dos usuários r e u; $\vec{r}(i)$ é a avaliação do item i realizada pelo usuário r; e $\vec{u}(i)$ é a avaliação do item i realizada pelo usuário u.

$$\varphi(r,u) = \sum_{i \in I} \sqrt{\left(\vec{r}(i) - \vec{u}(i)\right)^2} \tag{2.2}$$

2.2.3 Perfis Simbólicos

Esta técnica de filtragem utiliza o conceito de perfis simbólicos para medir a distância entre os perfis dos usuários, e assim poder apontar os mais semelhantes [16].

Diferentemente das medidas apresentadas nas seções 2.2.1 e 2.2.2, esta técnica adota uma estrutura simbólica para a construção dos perfis dos usuários, para logo depois, medir a similaridade entre eles. O processo de criação do perfil do usuário é feito em duas etapas: pré-processamento e generalização. A idéia central do pré-processamento é montar a estrutura simbólica para cada item que o usuário mostrou algum interesse. E a etapa de generalização cria de fato o perfil do usuário, que é formado por um conjunto de sub perfis. Cada sub perfil é composto de uma relação entre uma nota possível que pode ser dada pelo usuário e informações dos itens que foram analisados com esta mesma nota.

Após a criação dos perfis dos usuários, a vizinhança de um usuário alvo já pode ser encontrada para a continuação do algoritmo colaborativo. A **Equação 2.3**, mostra a função de similaridade que deve ser aplicada aos usuários r e u.

$$\psi(r,u) = \frac{1}{|D|}\sum_{g \in D}\left(1 - \phi\left(y_{r_g}, y_{u_g}\right)\right) \tag{2.3}$$

O conjunto D refere-se aos possíveis níveis de interesse do usuário sobre um determinado item. A função de dissimilaridade ϕ é responsável pela comparação dos sub perfis $\boldsymbol{y_{r_g}}$ *e* $\boldsymbol{y_{u_g}}$ **,** relacionados ao nível de interesse g, e pertencentes aos usuários $\boldsymbol{r}$ e $\boldsymbol{u}$, respectivamente.

Essa função de dissimilaridade é uma versão adaptada da distância euclidiana que leva em consideração as distribuições dos pesos dos perfis dos usuários.

$$\phi\left(y_{r_g}, y_{u_g}\right) = \sqrt{\sum_{g_k \in D}\left(W(r_g, g_k) - W(u_g, g_k)\right)^2} \tag{2.4}$$

Na **Equação 2.4**, $W(r_g, g_k)$ e $W(u_g, g_k)$, são as distribuições dos pesos para o sub perfil relacionado ao nível de interesse g_k, dos usuários $\boldsymbol{r}$ e $\boldsymbol{u}$, respectivamente.

Para exemplificar a criação de um perfil simbólico, considere as avaliações feitas por 4 usuários em um domínio de filmes. A **Tabela 1** ilustra as avaliações realizadas por esses usuários a 3 filmes diferentes. As notas dadas variam entre 1 e 5, sendo 1 a mais baixa e 5 a mais alta.

Tabela 1. Matriz de avaliações dos usuários.

	Rei Arthur	**X-Men**	**Rocky V**
Mateus	2	-	5
Geraldo	-	3	4
Luiz	-	-	4
Rafael	3	4	1

Dentro da fase de pré-processamento é criada a descrição simbólica dos filmes que foram avaliados na **Tabela 1**. Para realizar este procedimento é necessário extrair a distribuição de pesos das notas em relação ao filmes. Considere $L = \{g_1, g_2, \dots g_k\}$ como o conjunto dos possíveis níveis de interesse de um usuário acerca de um item; $g_k \in L$, a nota dada por um usuário $\boldsymbol{u}$ a um item $\boldsymbol{i}$; e ρ_i^k, o conjunto de usuários que mostraram algum interesse g_k para o item $\boldsymbol{i}$. A **Equação 2.5** representa o cálculo para obter esta distribuição, onde $|\rho_i^k|$ é a cardinalidade do conjunto ρ_i^k.

$$Q_{g_k}(i) = \frac{|\rho_i^k|}{\sum_{k=1}^{k} |\rho_i^k|} \tag{2.5}$$

A descrição simbólica de um item $\boldsymbol{i}$. é dada por $\eta_i = (L, q(i))$, onde $q(i)$ é o referente a distribuição dos pesos de todos os níveis de interesse. A **Tabela 2** apresenta as descrições modais para os filmes da **Tabela 1**.

Tabela 2. Descrições simbólicas dos filmes da Tabela 1.

	Descição Simbólica Modal
Rei Arthur	$\big(\{1,2,3,4,5\}, (0,0.5,0.5,0,0)\big)$
X-Men	$\big(\{1,2,3,4,5\}, (0,0,0.5,0.5,0)\big)$
Rocky V	$\big(\{1,2,3,4,5\}, (0.25,0,0,0.5,0.25\ )\big)$

A criação do perfil simbólico dos usuários está inserida na etapa de generalização. Como descrito anteriormente, um perfil simbólico para um usuário é composto por sub perfis relacionados às informações dos itens com a mesma avaliação feita por esse usuário. Seja u_{g_k} o sub perfil do usuário u para o nível de interesse g_k, a descrição simbólica deste sub perfil é dada por $\gamma(u_g) = \big(L, q(u_g)\big)$, onde $q(u_g)$ é a distribuição dos pesos referentes aos itens.

$$\mathrm{W}(g_k) = \frac{1}{|u_{g_k}|} \sum_{i \in u_{g_k}} \left(Q_{g_k}(i) \right) \tag{2.6}$$

A **Equação 2.6** descreve o cálculo para descobrir a distribuição dos pesos para o conjunto de itens de mesma nota, onde $|u_{g_k}|$ é a cardinalidade para o conjunto u_{g_k}. Com base nas equações citadas, o perfil simbólico de Mateus está representado na **Tabela 3**. Em [16] encontra-se outro exemplo de criação de um perfil simbólico modal para um conjunto de usuários.

Tabela 3. Perfil simbólico modal de Mateus.

	Descição Simbólica Modal
$\gamma(Mateus_1)$	$\left(\{1,2,3,4,5\}, (0,0,0,0,0)\right)$
$\gamma(Mateus_2)$	$\left(\{1,2,3,4,5\}, (0,0.5,0.5,0,0)\right)$
$\gamma(Mateus_3)$	$\left(\{1,2,3,4,5\}, (0,0,0,0,0)\right)$
$\gamma(Mateus_4)$	$\left(\{1,2,3,4,5\}, (0,0,0,0,0)\right)$
$\gamma(Mateus_5)$	$\left(\{1,2,3,4,5\}, (0.25,0,0,0.5,0.25)\right)$

2.2.4 Função de Utilidade

Seguindo as etapas do algoritmo de filtragem colaborativa, após calcular a similaridade entre os usuários, a etapa seguinte será predizer a nota de um item ou gerar uma lista personalizada de itens. Para isso será necessário a utilização de uma função que medirá a utilidade de um item $\boldsymbol{i}$ para algum usuário $\boldsymbol{u}$**.** Apenas os usuários que pertencem à vizinhança do usuário alvo, serão considerados nos cálculos efetuados pela função de utilidade.

A idéia básica da função de utilidade, como apresentada na **Equação 2.7**, é combinar as avaliações dos usuários mais próximos do usuário $\boldsymbol{u}$ com seus

respectivos valores de similaridade, para assim predizer a nota de um determinado item $\boldsymbol{i}$ [1].

$$\Pi(\mathcal{r}, i) = \overline{\vec{r}} + \frac{\sum_{k=1}^{h} \left(\vec{u}_k(i) - \overline{\vec{u}}_k\right) * \Phi(\mathcal{r}, u_k)}{\sum_{k=1}^{h} \Phi(\mathcal{r}, u_k)} \tag{2.7}$$

Na **Equação 2.7**, $\mathcal{r}$ é o usuário alvo para o qual a medida de relevância do item $\boldsymbol{i}$ será calculada; $\boldsymbol{h}$ é o número de vizinhos a ser considerado; $\boldsymbol{u_k}$ representa o $\boldsymbol{k}$-ézimo vizinho do usuário $\mathcal{r}$; $\overline{\vec{r}}$ e $\overline{\vec{u}}_{\boldsymbol{k}}$ representam as médias aritméticas das avaliações dos usuários $\mathcal{r}$ e $\boldsymbol{u_k}$, respectivamente; $\vec{\boldsymbol{u}}_{\boldsymbol{k}}(\boldsymbol{i})$ é a avaliação do usuário $\boldsymbol{u_k}$ para o item $\boldsymbol{i}$; e $\Phi(\mathcal{r}, u_k)$ o valor da similaridade entre o usuário $\boldsymbol{u_k}$ e o usuário $\mathcal{r}$.

Nota-se que independentemente do algoritmo de filtragem colaborativa que foi implementado, a função de utilidade usada nas predições é a mesma, pois todas as medidas descritas acima derivam de um algoritmo colaborativo padrão baseado no método kNN.

2.3 Qualidade das Recomendações

Há uma grande variedade de métricas utilizadas na literatura para avaliar a qualidade de um algoritmo de recomendação. A abundância de métricas dificulta para muitos pesquisadores, avaliar sistemas de recomendação e afirmar que um algoritmo é considerado melhor que o outro [20]. A questão é analisar que pontos do sistema de recomendação serão feitos a avaliação, para assim escolher devidamente uma métrica de avaliação.

Podem-se destacar na literatura, duas maneiras de se verificar a qualidade de uma recomendação:

i. Verificar a precisão de uma nota dada a um determinado item
ii. Verificar a precisão de uma lista de itens recomendados.

A primeira maneira verifica quão preciso um algoritmo de filtragem pode ser ao prever uma nota de um item específico. E segunda maneira, verifica a capacidade de um algoritmo de produzir uma lista de itens para um usuário alvo, ordenados pela relevância, como se esse próprio usuário tivesse feito manualmente

[10]. Desta forma os itens listados não são tratados de uma maneira binária, “bons” e “ruins”, mas de uma maneira que os apresente ordenados para o usuário por nível de interesse. Os “melhores” estarão nas primeiras posições e os itens restantes poderão ser considerados “bons” e “regulares”.

As análises comparativas que serão realizadas neste trabalho usarão a segunda maneira de avaliação de qualidade. Cada algoritmo de filtragem colaborativa implementado, será alvo dessas análises nos mais diferentes cenários dentro do ambiente experimental proposto.

2.3.1 Métrica de Breese

A métrica escolhida para medir a qualidade das recomendações é a criada por Breese [21]. Esta métrica tem a característica de através de uma lista de itens, calcular a utilidade final da recomendação baseando-se na utilidade de cada item descontado de um valor referente à posição deste item na lista.

Grande parte dos usuários que trafegam em sistemas web, quando requisitam uma busca sobre um determinado assunto, não costuma observar profundamente a lista retornada, no máximo os primeiros resultados. Sob esta visão que a métrica proposta por Breese é calculada. A métrica utiliza o conceito de *half-life,* que é a forma de representar o decaimento exponencial da utilidade do item na lista de itens através de uma constante $\boldsymbol{\alpha}$. Observa-se na **Equação 2.8**, que a partir do primeiro item da lista de itens ordenada, as chances de o usuário observar são de 50% em relação ao próximo item [22]. A utilidade de uma lista ordenada para o usuário $\boldsymbol{u}$ segundo a métrica de Breese será:

$$R_u = \sum_j \frac{\max\left(r_{u,j} - d, 0\right)}{2^{(j-1)/(\alpha-1)}} \tag{2.8}$$

Na **Equação 2.8**, $r_{u,j}$ é a avaliação dada pelo usuário $\boldsymbol{u}$ ao item que se encontra na posição j da lista ordenada pelo sistema. A variável d refere-se ao valor médio no intervalo de avaliação ou um determinado valor limiar. A utilidade do item será a diferença entre a avaliação e o valor de d. A constante $\boldsymbol{\alpha}$ foi utilizada por Breese em seus experimentos, com o valor igual a 5. Observou-se também que o uso da constante *half-life* igual a 10 causou pouca sensibilidade nos resultados [10].

Os experimentos propostos utilizaram um *half-life* igual a 5. Assim o valor final da métrica para um conjunto ***n*** usuários é dado por:

$$R = 100\ \frac{\sum_n R_n}{\sum_n R_n^{max}} \tag{2.9}$$

Na **Equação 2.9**, R_n^{max} é a utilidade máxima obtida quando todos os itens avaliados pelo usuário R_n foram ordenados. Note que o valor de R varia entre 0, para uma utilidade nula, e 100, para uma utilidade máxima alcançada pelo sistema. Caso o valor do quociente $\frac{R_u}{R_u^{max}}$ for igual a 1, isto significa que a técnica de filtragem usado pelo sistema, conseguiu ordenar o conjunto de itens em questão com a mesma relevância dada pelo usuário $\boldsymbol{u}$.

2.4 Abordagem Inter-aplicações

Os sistemas de recomendação atuais utilizam o perfil do usuário obtido em sua aplicação para realizar recomendações personalizadas. O usuário por sua vez, acessa diversas aplicações e navega por elas interagindo com os mais diferentes tipos de conteúdo. Cada aplicação que trabalha com um engenho de recomendação, possui um perfil para cada usuário. Todas essas aplicações limitam-se em ter uma única visão dos seus usuários para indicar produtos de seu conjunto de itens. Em cima deste ponto que um sistema que utiliza uma abordagem inter-aplicações realiza melhorias nas suas recomendações.

2.4.1 Conceito Geral

A proposta de uma abordagem inter-aplicações é unir os perfis de usuários que estão ligados a várias aplicações e assim agregar uma maior quantidade de informações acerca deles. Com esse conhecimento adquirido, o sistema que utiliza esta interface de integração inter-aplicações, poderá fornecer recomendações mais precisas e interessantes, além de atenuar alguns problemas que técnicas de filtragem colaborativa apresentam na geração de recomendações. Detalhes de uma implementação deste tipo de sistema encontra-se em [4].

2.4.2 Características

Um sistema de recomendação inter-aplicações deve se preocupar principalmente com os seguintes pontos:

i. Manipular diferentes perfis de diversas aplicações
ii. Fornecer um baixo nível de acoplamento
iii. Possuir interoperabilidade

O primeiro ponto refere-se à capacidade do sistema em manipular os mais diferentes perfis de modo a conseguir realizar recomendações a qualquer aplicação que se utilize dele. O modo como o usuário realiza as avaliações varia entre as diferentes aplicações. Por exemplo, em uma aplicação em que o usuário se depara com um conjunto de bandas ou artistas, as avaliações serão feitas usando os indicadores “gosto” ou “não gosto”. E em outra aplicação, por exemplo, o usuário expressará seu gosto sobre uma lista de filmes, utilizando os indicadores: “Excelente”, “Muito Bom”, “Bom”, “Regular” e “Ruim”. Assim o sistema de recomendação inter-aplicações precisa ter uma técnica de personalização de conteúdo bem definida para conseguir trabalhar com mais de uma aplicação.

O segundo e o terceiro ponto entram no mérito de como o sistema inter-aplicações deverá ser modelado e implementado para fornecer as devidas recomendações. Uma alternativa é utilizar uma interface de integração na forma de um serviço, que será acessível a todas as aplicações que desejam usá-la. Como em [4], foi utilizado no ambiente experimental deste trabalho uma implementação baseada em *Web Service*. As vantagens de se utilizar uma arquitetura orientada a serviços serão detalhadas na próxima seção.

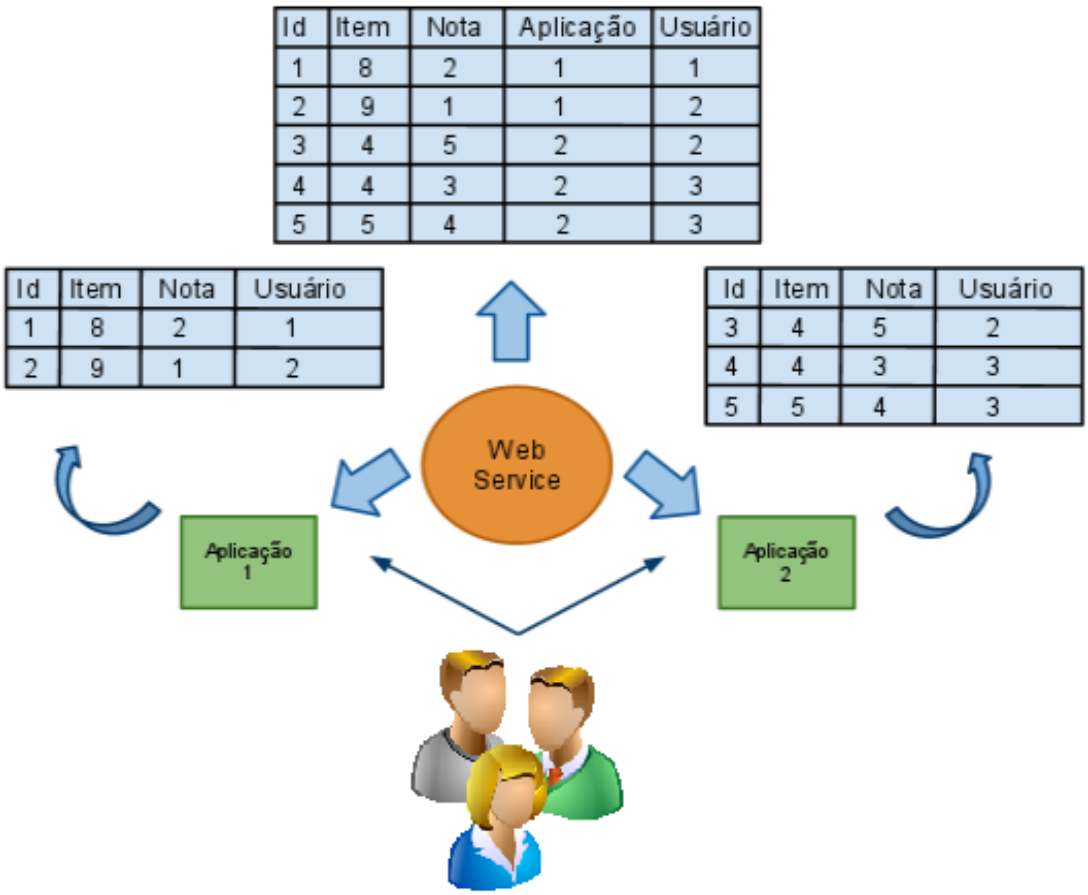

Figura 4. Exemplo de um ambiente inter-aplicações.

A **Figura 4** apresenta um ambiente onde duas aplicações utilizam o serviço de integração inter-aplicações. Para realizar recomendações para mais de uma aplicação, a lógica implementada pelo serviço deverá seguir alguns critérios pré-estabelecidos. Os critérios descritos abaixo estão relacionados aos cálculos dos vizinhos mais próximos e ao cálculo da função de utilidade:

- Um usuário só poderá ser incluso nos cálculos de vizinhança, se ele possuir ao menos uma preferência dentro da aplicação que invocou o serviço de integração inter-aplicações.
- O usuário que utilizar o serviço a partir de uma aplicação não precisará possuir alguma avaliação nesta aplicação, mas deverá possuir alguma avaliação em outra aplicação que também utilize o serviço de integração.
- Cálculos envolvendo predições de nota para um determinado item ou recomendação de uma lista de itens, sempre considerarão os itens pertencentes à aplicação que invocou o serviço.

Os pontos descritos acima devem ser levados em consideração nos cálculos que são feitos dentro do serviço de integração inter-aplicações para evitar possíveis inconsistências nas recomendações geradas.

2.4.3 Arquitetura Orientada a Serviços (SOA) e Web Services

Ao longo do tempo, negócios na área de TI ficaram mais dinâmicos e mais competitivos. As empresas especializadas em TI se depararam com problemas de integração e comunicação entre sistemas heterogêneos, necessidade de rapidez a mudanças de requisitos, e também a necessidade de criar aplicativos reusáveis mais dinâmicos e menos acoplados. Uma saída para esses problemas foi à criação de um tipo de arquitetura orientada a serviços (SOA) [23].

O serviço pode ser visto como uma unidade lógica executável usada no paradigma de orientação a serviços. Uma de suas características é ser encapsulado, permitindo assim seu acesso somente através de uma interface sólida que inclua um padrão de interação. Geralmente os serviços são fornecidos para a realização de uma atividade de negócios específica dentro de um dado contexto [24, 25].

Web Services são tipos de serviços altamente integrados com a arquitetura orientada a serviços, que possuem uma abordagem de computação distribuída e capacidade de integração com diferentes aplicativos através da internet. Os *Web Services* são completamente independentes de plataforma, infra-estrutura e linguagem de programação, fornecendo um baixo acoplamento entre o fornecedor e o consumidor do serviço. Esse tipo de serviço utiliza tecnologias de padrão aberto como: XML(do inglês, eXtensible Markup Language), WSDL (do inglês *Web Service Definition Language*) e SOAP (do inglês *Simple Object Access Protocol*).

XML é um padrão de representação de dados baseado em texto, que permite definir uma gramática para descrever praticamente qualquer tipo de dados. WSDL é um tipo de documento baseado em XML, que descreve as mensagens que o *Web Service* pode aceitar e toda sua estrutura de respostas. SOAP é um protocolo de transporte de dados baseado em XML que fornece estruturas para descrição de conteúdo e processos. A **Figura 5** ilustra a troca de informações entre o cliente e o serviço através das mensagens SOAP. Mais detalhes sobre essas tecnologias podem ser encontrados em [25, 26].

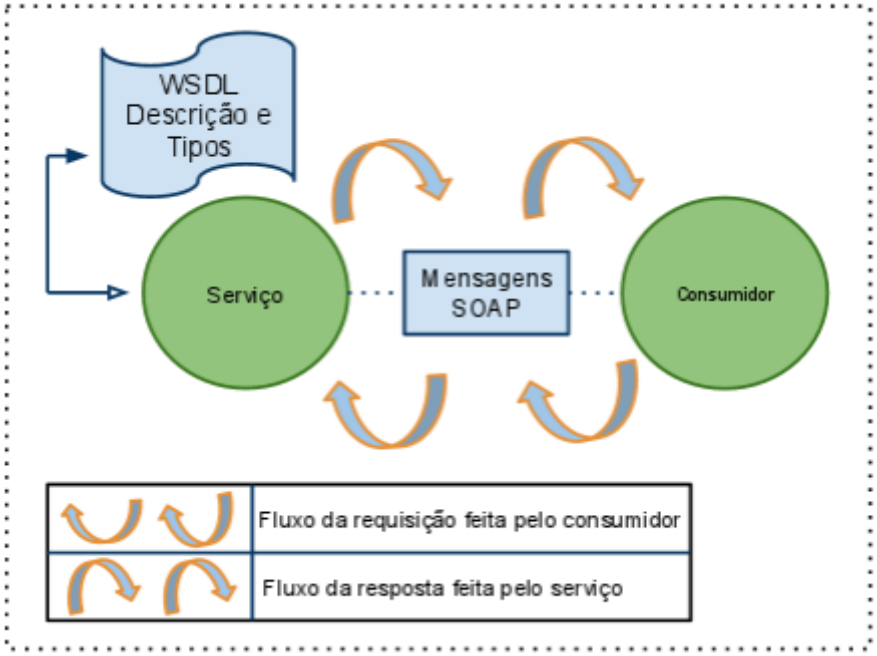

Figura 5. Comunicação entre o Web Service e o cliente da aplicação.

A partir desses conceitos podem-se listar algumas características que levaram a escolha da arquitetura orientada a serviços como a adotada para o desenvolvimento de sistemas de recomendação inter-aplicações [24, 25]:

i. Maior potencial de reutilização.
ii. Transações podem ser processadas separadamente.
iii. É uma solução mais escalável.
iv. Fácil integração com outros sistemas.
v. Facilidade de implementação.

Com a utilização deste tipo de serviço junto com a arquitetura apresentada pode-se agregar vários ganhos para o sistema de recomendação como os citados acima, além de tentar melhorar a qualidade das recomendações geradas pela aplicação e atenuar alguns problemas das técnicas de filtragem colaborativa.

Capítulo 3

Ambiente Experimental

Neste capítulo estão descritos os detalhes do desenvolvimento do sistema de recomendação que foi usado para realizar e analisar as recomendações. Na seção 3.1 estão alguns pontos da implementação do serviço de integração inter-aplicações, além de uma visão geral de algumas estruturas do ambiente desenvolvido. Por último, na seção 3.2 estão apresentadas as tabelas usadas na modelagem do banco de dados e detalhes da carga feita na base para a realização dos experimentos.

3.1 Sistema Desenvolvido

Para realizar a análise das técnicas de filtragem colaborativa, foi necessária a construção de um ambiente para suportar recomendações tanto tradicionais como aquelas baseadas em um cenário inter-aplicações, considerando os conceitos que foram introduzidos no capítulo 2. O sistema proposto foi desenvolvido com o intuito de servir como base para os experimentos deste trabalho e como referência para trabalhos futuros.

3.1.1 Visão Geral

O sistema foi desenvolvido como uma aplicação web que será responsável pela geração e análise das recomendações. As funcionalidades básicas desta aplicação são: gerar uma lista de itens personalizada para um usuário alvo e obter o valor da métrica Breese (seção 2.3.1) para um conjunto determinado de usuários que estejam inseridos no sistema. Os itens utilizados nas recomendações pertencem a três conjuntos de gêneros diferentes. Um conjunto está relacionado a filmes, outro no contexto de bandas e cantores, e o outro no âmbito dos livros.

A aplicação fornece uma interface simples que será usada para realizar as ações específicas de recomendação e análise. A **Figura 6** ilustra o menu da aplicação desenvolvida, onde podem ser observadas três abas principais:

- *Users*
- *Recommendation*
- *Analysis of Recommendations*

Figura 6. Menu do ambiente experimental desenvolvido.

Ao selecionar a aba de usuários, serão listados os usuários que estão cadastrados no sistema. Todos esses usuários serão passíveis de avaliações sobre qualquer contexto de itens e de futuras recomendações e análises. Nesta mesma aba, estará disponível um *link* que redirecionará a aplicação para uma área onde poderá ser cadastrado um novo usuário no sistema.

Ao selecionar de recomendações, estarão visíveis quatro campos que serão utilizados como parâmetros de entrada para a geração da lista de recomendações. Além disso, esta aba fornece uma lista dos usuários mais próximos do usuário alvo e uma lista dos itens que já foram previamente avaliados por ele. Estas duas listas citadas serão listadas na tabela *Neighborhood*, para a vizinhança do usuário alvo, e na tabela *Rated Items*, para os itens que já foram avaliados. Os campos que servirão para parâmetros das recomendações são:

i. *User*
ii. *Technique*
iii. *Application Type*
iv. *Inter-application approach*

O primeiro campo deverá ser preenchido com o email do usuário alvo que se deseja recomendar uma lista de itens. No segundo campo estarão listadas as três técnicas de filtragem colaborativa que foram implementadas neste trabalho. No terceiro campo estarão listadas os três tipo de aplicações usadas nas avaliações dos usuários e nas recomendações inter-aplicações. E o último campo indica se as recomendações vão considerar apenas a aplicação alvo ou todas as outras aplicações. Os itens recomendados pela aplicação serão listados na tabela *Recommended Items*. **A Figura 7** ilustra a área de recomendações.

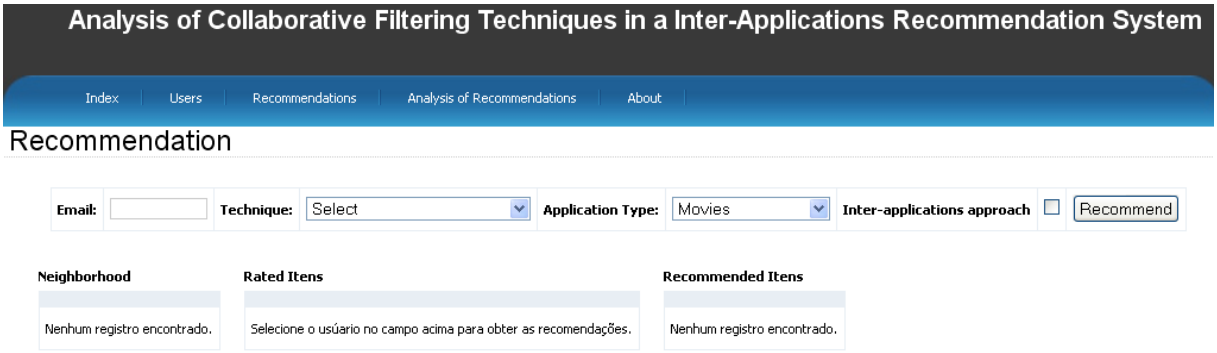

Figura 7. Tela de recomendações**.**

Ao selecionar a aba de análise das recomendações, estarão disponíveis todos os fatores que influenciarão as recomendações e que serão responsáveis pela formação dos diferentes cenários propostos neste trabalho. Os campos utilizados para montar os cenários foram:

i. Database Size
ii. Number of items rated by user
iii. Number of neighbors
iv. Technique
v. Application Type
vi. Inter-application approach

O primeiro campo está relacionado com a quantidade de usuários disponíveis na base de dados no momento da geração das recomendações. O segundo campo deve ser preenchido com a quantidade de itens avaliados, que serão considerados para cada usuário na construção das recomendações. O terceiro campo define a quantidade de vizinhos mais próximos (o valor de k) para cada usuário. O quarto campo listará as técnicas de filtragem colaborativa desenvolvidas para a análise das recomendações. No quinto campo, como na aba anterior, estarão listados os três tipos de aplicações envolvidas na análise. E o último campo indicará se a análise considerará apenas a aplicação alvo ou o restante das aplicações. Logo abaixo à tabela de parâmetros, encontra-se um campo que mostrará o valor calculado da métrica Breese para o cenário criado. A **Figura 8** ilustra a área de análise das recomendações descrita acima.

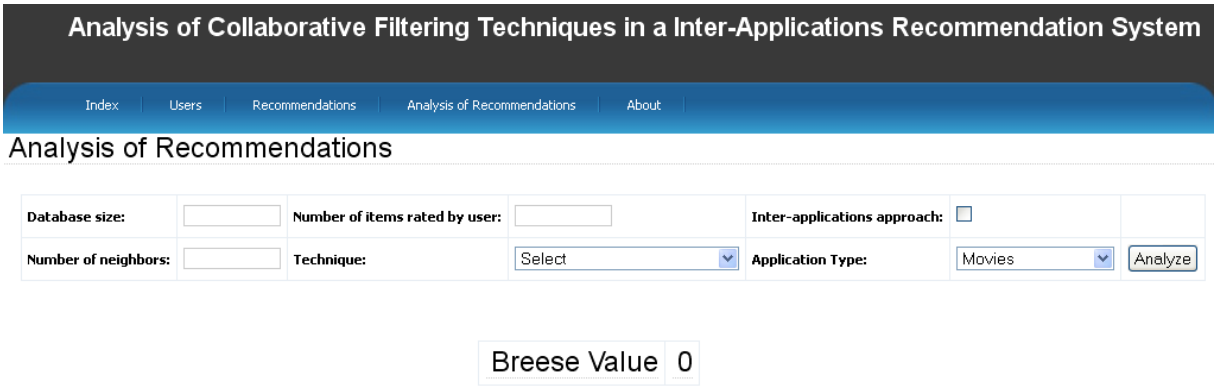

Figura 8. Tela de análise das recomendações.

Os detalhes de como esses campos foram utilizados nas análises serão descritos no próximo capítulo.

3.1.2 Tecnologias

O ambiente de recomendações foi desenvolvido utilizando a plataforma .NET da Microsoft na versão 4.0, através da ferramenta de desenvolvimento *Visual Web Developer 2010 Express*[1]. A linguagem C# foi usada na codificação de todo o sistema juntamente com tecnologias específicas para desenvolvimento de aplicações web e desenvolvimento de aplicativos orientados a serviço. O servidor de banco de dados utilizado foi o *SQL Server 2008 R2 Express*[2].

Toda a estrutura da aplicação web foi estabelecida utilizando o *framework ASP.NET MVC 2*[3]. Este *framework* fornece ferramentas para a criação de aplicativos web robustos e integrados com os recursos da plataforma .NET, usando o padrão arquitetural MVC (do inglês, *Model-View-Controller*). O WCF (do inglês, *Windows Communication Foundation*) é o *framework* desenvolvido pela Microsoft que unificou tecnologias de computação distribuída em uma única solução, baseando-se no paradigma de orientação a serviços. Este *framework* foi usado na criação do serviço de integração inter-aplicações.

[1] www.microsoft.com/express/Web

[2] www.microsoft.com/express/Database

[3] www.asp.net/mvc

A escolha das tecnologias para todo o desenvolvimento da aplicação foi feita pelo domínio dessa plataforma por parte do desenvolvedor, além das vantagens que essa plataforma fornece tais quais: robustez, dinamismo e facilidade de desenvolvimento.

3.1.3 Arquitetura

A aplicação web foi estruturada sobre o padrão arquitetural MVC juntamente com os conceitos de orientação a serviços. Basicamente a aplicação pode ser dividida em duas camadas principais: a camada de apresentação e a camada de serviço. A camada de apresentação abrange toda a parte de desenvolvimento web, regras de negócios, camada de acesso a dados e o cliente para o serviço de recomendação inter-aplicações. A camada de serviço abrange toda a implementação do serviço de recomendação, acesso a dados e a interface de contrato. A **Figura 9** ilustra como a aplicação foi projetada.

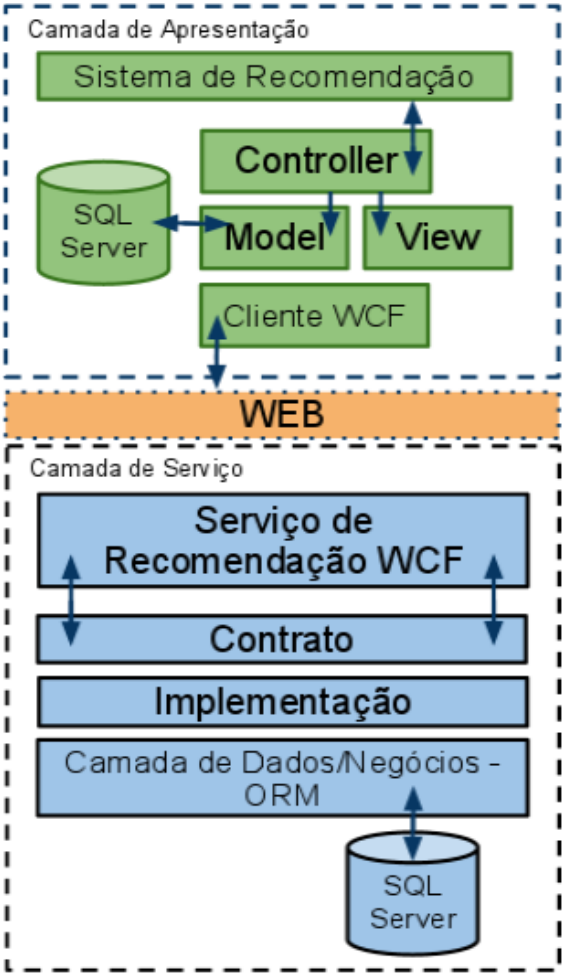

Figura 9. Arquitetura adotada no desenvolvimento do ambiente experimental.

A camada de apresentação é o ponto de entrada no sistema para que as ações propostas por este trabalho sejam realizadas. A manipulação da requisição é feita pela estrutura *controller*. Esta estrutura está integrada com todas as regras de negócios e com os modelos existentes no sistema. O *controller* também será responsável por executar qualquer requisição envolvendo operações de recomendação ou análise através do cliente do serviço de recomendação situado na aplicação web.

A camada de serviço representa a estrutura que receberá requisições e fornecerá como respostas as recomendações desejadas ou alguma análise de um cenário criado. O serviço conta com uma interface, chamada de contrato, que representa o ponto de integração entre o consumidor e fornecedor das operações envolvidas. A implementação do serviço foi através de um *Web Service*, utilizando o WCF para o suporte de toda a infra-estrutura de orientação a serviços. Para a realização do estudo foram necessárias apenas três ações específicas presentes no contrato e ilustradas na **Figura 10**.

Figura 10. Operações presentes no contrato do serviço de integração inter-aplicações.

O objetivo da operação *RecommendItemsList* é recomendar uma lista de itens personalizados para um usuário alvo com base nas seguintes entradas: o tipo de aplicação que será feita a recomendação, o usuário alvo, o número ***k*** de vizinhos, se vai ser uma recomendação tradicional ou inter-aplicações, e a técnica

de filtragem colaborativa. Todos esses parâmetros de entrada são obtidos através da interface gráfica que foi previamente apresentada.

A operação *GetNearestNeighbors* tem a função de encontrar os vizinhos mais próximos de um usuário alvo. Os parâmetros usados para realizar esta operação são: o tipo de aplicação, o identificador do usuário no sistema, se vai ser utilizada a abordagem tradicional ou inter-aplicações, e a técnica de filtragem colaborativa.

E a última operação é a *CalculateRecommendatiosQuality*, que é responsável por calcular o valor da métrica Breese de acordo com as entradas a seguir: número de usuários levados em consideração, número de itens avaliados pelos usuários, se vai ou não considerar outras aplicações, o tipo de técnica de filtragem colaborativa escolhida, quantidade de vizinhos mais próximos e a aplicação que invocou a operação. Como na primeira operação do contrato, esses parâmetros serão passados através da interface gráfica do ambiente experimental.

Após a definição do contrato, a implementação foi baseada em três interfaces principais. Duas delas usadas para o desenvolvimento das operações de recomendação e uma para a operação de análise. A **Figura 11** mostra as interfaces usadas.

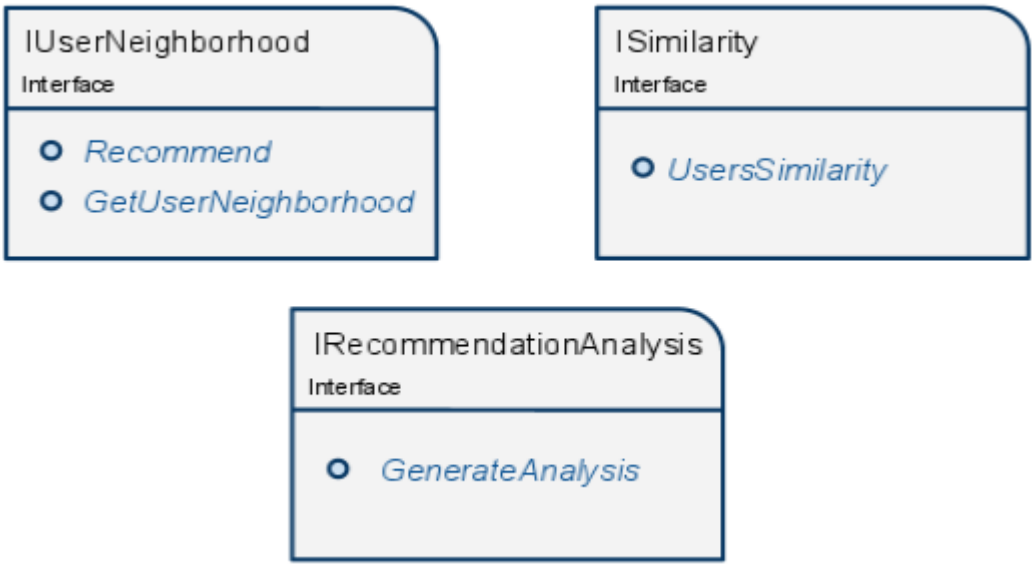

Figura 11. Interfaces usadas dentro do serviço de integração inter-aplicações.

A interface *IUserNeighborhood* define como as técnicas de filtragem colaborativa implementadas devem se comportar. As técnicas precisam recomendar

uma lista de itens ao usuário alvo e identificar os usuários mais próximos a ele. A interface *ISimilarity* define como as funções de similaridade devem responder. Esta interface é implementada pelos três algoritmos propostos que retornam o valor representado pela similaridade entre dois usuários. E a interface *IRecommendationAnalysis* corresponde ao comportamento das análises de qualidade das recomendações. A métrica Breese foi implementada em cima desta interface para a análise das listas de itens sugeridas pelo sistema.

3.2 Base de Dados

Como citado anteriormente, o repositório de conhecimento é composto por conjuntos de itens separados em três domínios diferentes. Esses itens foram utilizados para serem avaliados pelos usuários inseridos no sistema e para o auxilio na construção de um ambiente experimental de recomendações. Os três tipos de itens usados foram:

- Filmes
- Bandas e Cantores
- Livros

No conjunto de filmes foram usados 30 itens, que serviram para as avaliações dos usuários. As notas dadas se restringem a um valor entre 1 e 10. Desta forma o usuário poderá avaliar o filme em dez níveis de interesse. Foram usados no conjunto de bandas e cantores 30 itens. As preferências foram feitas com os dois valores booleanos, “gosto” ou “não gosto”. E o último conjunto, foi formado por 30 livros para que os usuários demonstrassem suas preferências. Os níveis de interesses para o grupo de livros foram valores entre 1 e 5. Esses valores retratam as opiniões dos usuários na forma de “ruim”, “regular”, “bom”, “muito bom” e “excelente”. Os três contextos apresentados representam as aplicações que utilizam o serviço de integração para a simulação de um ambiente inter-aplicações.

Foi utilizado nos experimentos o número de 60 usuários. Todos esses usuários avaliaram nos contextos descritos pelo menos 10 itens de cada, portanto cada usuário terá no mínimo 30 avaliações no sistema. Ao todo foram utilizadas

2297 avaliações de usuários nos três conjuntos de itens para realizar as recomendações e análise do sistema de recomendação inter-aplicações.

A base de dados foi estruturada usando as seguintes tabelas:

i. Usuários
ii. Itens
iii. Preferências
iv. Contextos das Aplicações
v. Itens das Aplicações

Esta modelagem foi adotada para representar os elementos necessários de um sistema de recomendação inter-aplicações A **Figura 12** mostra o modelo de dados relacional do sistema projetado para execução dos experimentos

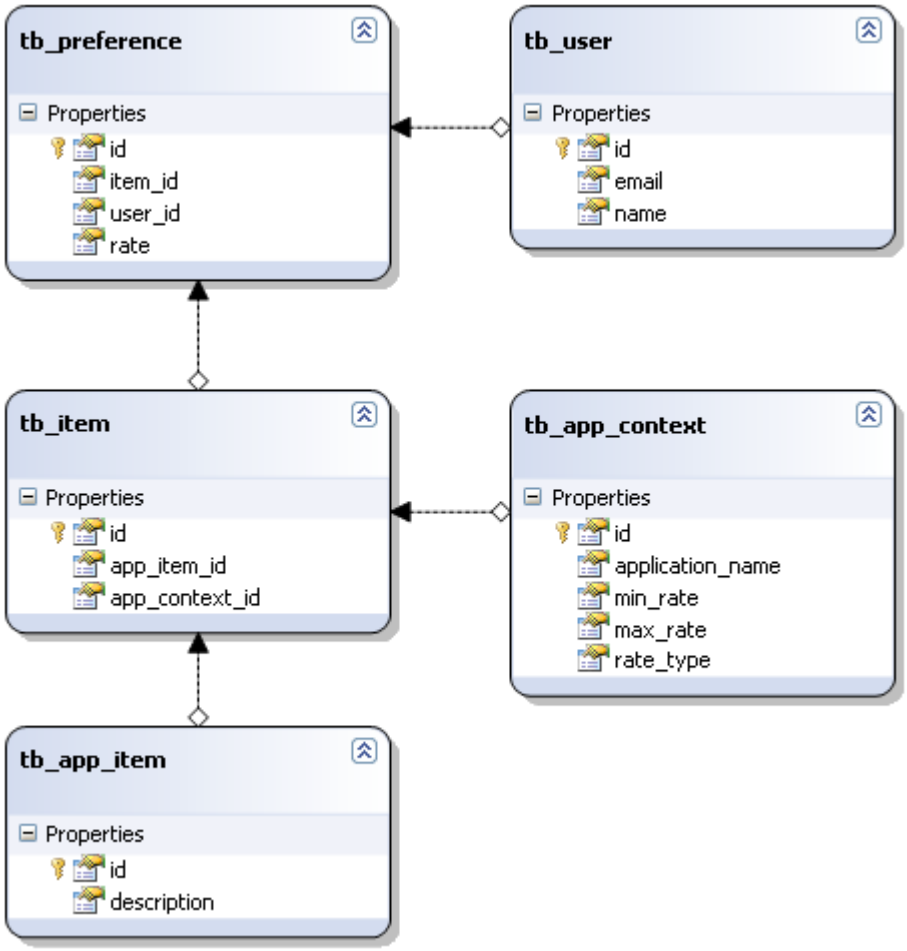

Figura 12. Diagrama do banco de dados usado nos experimentos.

A tabela *tb_user* armazena os usuários que foram inseridos no sistema e estão aptos a realizar alguma avaliação. Esta tabela apresenta os seguintes campos: *id, email* e *name*. O primeiro campo representa o identificador do usuário e chave primária no sistema de recomendação inter-aplicações. O segundo representa o email e *login* do usuário. E o terceiro campo, o nome do usuário cadastrado.

A tabela *tb_app_context* armazena os gêneros dos itens das aplicações que usam o sistema de recomendação inter-aplicações. O campo *id* é o identificador do contexto no sistema e também é a chave primária. O campo *application_name* representa o nome da aplicação que vai utilizar o sistema. Os campos *min_rate* e *max_rate* representam a menor e maior nota que se pode avaliar um item, respectivamente, nesta aplicação. O campo *rate_type* está relacionado com o tipo numérico que a nota está associada.

A tabela *tb_app_item* é uma forma de representar o conjunto de itens de todas as aplicações que utilizam o serviço de integração inter-aplicações. Nesta tabela estarão os itens referentes aos filmes, artistas e livros. O campo *id* indica o identificador desse item na aplicação que ele pertence e também sua chave primária. O campo *description* representa o rótulo do item (ex: “Rei Arthur”).

A tabela *tb_item* representa o conjunto de itens que de fato são usados no sistema de recomendação desenvolvido. Esta tabela associa um item da tabela *tb_app_item* com um contexto da tabela *tb_app_context*. O campo *id* é o identificador e chave primária do item no sistema de recomendação inter-aplicações. Os campos *app_item_id* e *app_context_id* são respectivamente, a referência do item que está presente na tabela *tb_app_item* e a referência do contexto de aplicação presente na tabela *tb_app_context*.

Na tabela *tb_preference* estão todas as preferências que envolvem os usuários e os itens armazenados no sistema desenvolvido. O campo *id* é o identificador e chave primária da preferência no sistema de recomendação inter-aplicações. O campo *user_id* é o identificador do usuário no sistema e o campo *item_id* o identificador do item no sistema. O campo *rate* representa a nota dada pelo usuário ao item em questão, respeitando o contexto de aplicação do item.

Toda essa estrutura apresentada tenta simular uma base de dados de um sistema de recomendação que utiliza também, recomendações inter-aplicações. Por

não existir ainda um serviço real tal que consiga integrar aplicações para sugerir itens, foi necessária a criação e carga de toda a base de dados usada nos experimentos. Para isso, foi feita uma coleta que contou com a participação, dentre outras pessoas, de estudantes e professores do curso de Engenharia de Computação da Escola Politécnica de Pernambuco – UPE, que expressaram suas opiniões sobre itens presentes nas três fontes de dados usadas nesta simulação.

Capítulo 4

Análise das Recomendações

Os experimentos e resultados descritos neste capítulo foram obtidos a partir do ambiente experimental desenvolvido no capítulo anterior. Na seção 4.1 será apresentada a metodologia usada para a realização dos experimentos. E na seção seguinte, todos os cenários e experimentos propostos por este trabalho para a análise de qualidade de um sistema de recomendação inter-aplicações.

4.1 Estrutura da Análise

4.1.1 Configuração dos Cenários

Dentro do ambiente experimental foram criados vários cenários para representar diferentes ambientes que podem ser encontrados em um sistema de recomendação. Para que isso ocorra, alguns pontos do sistema foram permutados para simular esses novos ambientes. Os fatores levados em consideração para a construção dos cenários foram:

- Número de usuários na base de dados
- Número de itens avaliados pelos usuários
- Recomendações usando abordagem inter-aplicações ou tradicional
- Tipo de algoritmo de filtragem colaborativa

O número de usuários utilizados nos experimentos vai indicar a quantidade de usuários presente na base de dados no momento em que a recomendação foi feita. Foram criados três tipos de tamanho para a análise das recomendações: pequena, para 20 usuários; média, para 40 usuários; e grande, para 60 usuários. A quantidade de itens usados e avaliados pelos usuários foi especificada no capítulo anterior.

O número de itens avaliados pelos usuários reflete o quanto o sistema de recomendação conhece sobre o usuário em questão. Para verificar a sensibilidade do sistema para o problema do novo usuário dentre outros pontos, foram usadas três quantidades de itens para cada usuário testado no experimento $\{2, 5, 8\}$. Essas

quantidades foram escolhidas de forma proporcional a base de dados utilizada nos experimentos, além de indicar cenários reais em que o usuário, na maioria das vezes, não está disposto a responder várias questões sugeridas pelo sistema.

Recomendar itens aos usuários de maneira tradicional ou inter-aplicações, pode ser um fator interessante para verificar o comportamento do sistema de recomendação em relação ao problema do novo usuário e ao problema da ovelha negra. Este fator será um indicador se as recomendações geradas utilizaram ou não a abordagem inter-aplicações.

O tipo de algoritmo de filtragem colaborativa indica qual das técnicas implementadas foi utilizada para sugerir uma lista personalizada de itens. Os algoritmos envolvidos nas análises foram aqueles apresentados na seção 2.2, ou seja: coeficiente de correlação de Pearson (seção 2.2.1), distância euclidiana (seção 2.2.2) e em perfis simbólicos usando FC (seção 2.2.3). Todos os outros fatores citados acima influenciaram diretamente o desempenho das técnicas de filtragem que utilizaram esses algoritmos. Desta forma, a avaliação das técnicas vem a ser o ponto principal do trabalho proposto.

Cada técnica foi submetida a um cenário específico, onde foram analisadas as diferentes respostas obtidas por cada uma através da métrica Breese. A **Tabela 4** mostra todos os cenários montados para os experimentos propostos. As avaliações ocorreram separadamente para cada cenário descrito, mediante a uma técnica de filtragem colaborativa.

Tabela 4. Os cenários envolvidos nas análises.

	Tamanho da Base de dados	*Nº de itens avaliados*
Cenário 1	Pequena (20 usuários)	2
Cenário 2	Pequena (20 usuários)	5
Cenário 3	Pequena (20 usuários)	8
Cenário 4	Média (40 usuários)	2
Cenário 5	Média (40 usuários)	5
Cenário 6	Média (40 usuários)	8
Cenário 7	Grande (60 usuários)	2
Cenário 8	Grande (60 usuários)	5
Cenário 9	Grande (60 usuários)	8

A nomenclatura utilizada nas tabelas de resultados foi baseada na variável R_h^t e exemplificada na **Tabela 5**. As técnicas de filtragem colaborativa representada pela variável t, foram escolhidas da seguinte maneira: ps, para perfis simbólicos; de, para distância euclidiana; e ccp, para coeficiente de correlação de Pearson. O número de vizinhos mais próximos (valor de k) é representado pela variável h e pertence ao conjunto $\{5, 10\}$.

Tabela 5. Nomenclatura usada nos experimentos de análise.

	Técnica de FC	***Nº de Vizinhos***
$\boldsymbol{R_5^{ps}}$	Perfis simbólicos	5
$\boldsymbol{R_{10}^{ps}}$	Perfis simbólicos	10
$\boldsymbol{R_5^{de}}$	Distância Euclidiana	5
$\boldsymbol{R_{10}^{de}}$	Distância Euclidiana	10
$\boldsymbol{R_5^{ccp}}$	Coef. de Correlação de Pearson	5
$\boldsymbol{R_{10}^{ccp}}$	Coef. de Correlação de Pearson	10

4.1.2 Procedimento de Execução

Após a descrição de como os cenários foram compostos, será dada a visão dos experimentos relatando os procedimentos necessários para a obtenção do valor da métrica Breese.

A análise das recomendações é iniciada pela interface gráfica que fornece todos os parâmetros para a configuração de um cenário. A escolha de quais usuários fará parte do conjunto de usuários escolhidos tendo em vista o tamanho especificado da base de dados, se deu de maneira aleatória a partir de toda a base disponível. Também foi escolhido aleatoriamente o perfil do usuário alvo, baseado no parâmetro número de itens avaliados pelo usuário. Com o restante dos itens avaliados por esse usuário alvo, foi criado um conjunto de itens de teste que servirão para a avaliar a capacidade dos algoritmos em sugerir os itens, segundo a métrica Breese. Por trabalhar com uma base fixa, a medida que o parâmetro número de

itens avaliados pelo usuário cresce, o conjunto de teste tende a diminuir acarretando em um quantidade reduzida de itens para a avaliação dos algoritmos de filtragem.

Para obter a métrica Breese final de um cenário proposto, calculou-se trinta vezes para cada usuário pertencente ao conjunto escolhido. Desta forma, para cada cenário montado calculamos a métrica usando diferentes perfis de usuários, diferentes usuários compondo a base de dados e diferentes conjuntos de teste. A média foi calculada para cada usuário, para assim obter o valor final da métrica Breese.

4.2 Experimentos e Resultados

Os experimentos propostos envolvem a análise de qualidade das recomendações geradas pelas diferentes técnicas através do ambiente experimental. Foi dividido primeiramente em duas análises, uma em um ambiente de recomendação tradicional e o outro usando uma abordagem inter-aplicações. Estas duas análises envolvem todas as variações de cenários citadas anteriormente.

4.2.1 Análise de técnicas de filtragem colaborativa utilizando abordagem tradicional

O primeiro experimento pretende avaliar as três técnicas de filtragem colaborativa em um ambiente utilizando apenas uma aplicação. Os cenários descritos na **Tabela 4** foram usados para realizar as recomendações e posteriormente a análise de qualidade. Todas as citações com relação a numeração de algum cenário pode ser vista nessa tabela.

Para medir a qualidade das recomendações em um sistema de recomendação tradicional, foi usada a base de dados referente aos itens do contexto de filmes. Tendo em vista a quantidade de avaliações feitas pelos usuários, foi mais interessante a escolha do contexto de filmes, pois o número de opiniões envolvendo este contexto é maior em relação aos outros contextos. A **Tabela 6**, mostra os resultados da métrica Breese para o conjunto de itens da categoria de filmes.

Tabela 6. Resultados referentes aos valores da métrica de Breese para um sistema de recomendação utilizando a abordagem tradicional.

	Pequena			***Média***			***Grande***		
	8	*5*	*2*	*8*	*5*	*2*	*8*	*5*	*2*
R_5^{ps}	82,45	81,61	**81,77**	82,05	82,14	83,45	81,63	81,25	**82,41**
R_{10}^{ps}	79,26	**82,59**	80,22	83,12	**83,78**	**83,59**	80,27	80,67	80,78
R_5^{de}	84,51	79,62	73,73	85,92	80,59	75,33	86,3	80,62	75,31
R_{10}^{de}	82,3	81,06	76,14	**86,04**	81,82	75,55	**87,52**	80,01	76,48
R_5^{ccp}	**84,85**	80,37	70,36	83,99	80,34	74,63	85,48	81,08	75,12
R_{10}^{ccp}	83,28	80,27	71,78	84,27	81,97	75,18	86,11	**81,54**	76,05

Pode-se perceber que os valores das médias da métrica Breese, em geral, foram relativamente altos. Isso se deve ao número de itens usados no conjunto de teste que ficou em torno de 10 a 15 itens para a avaliação. Mas o importante é observar o desempenho dos algoritmos e todos os padrões que eles apresentaram neste experimento.

Começando as análises, observa-se que a técnica de filtragem que utilizou o algoritmo baseado em perfis simbólicos apresentou um ótimo resultado para cenários em que a quantidade de itens no perfil do usuário é pequena. A característica principal deste algoritmo é conseguir lidar com pouca informação acerca do usuário, e sugerir itens de uma maneira mais precisa. Foi comprovada esta característica nos primeiros cenários, principalmente nas configurações de tamanho de base de dados pequeno e médio.

As técnicas de filtragem colaborativa que foram implementadas usando os algoritmos baseados no coeficiente de correlação de Pearson e distância euclidiana, tiveram desempenhos similares dentro dos cenários de base de dados de mesmo tamanho. Estes dois algoritmos foram usados na forma clássica de um algoritmo kNN de filtragem. É possível perceber que em vários pontos o algoritmo baseado na distância euclidiana apresentou um desempenho melhor do que o algoritmo baseado na correlação de Pearson.

Um ponto que pôde ser visto nos resultados apresentados, foi que a mudança no número de vizinhos nos algoritmos baseados no coeficiente de correlação de Pearson e na distância euclidina, não afetaram de forma significativa os resultados em um mesmo cenário. Este fato pode ser mais observado no algoritmo de Pearson. Em alguns casos, como em $\boldsymbol{R_5^{ccp}}$ e $\boldsymbol{R_{10}^{ccp}}$ no cenário 2, o resultado ficou praticamente invariável. A **Figura 13** ilustra os resultados similares gerados pelo ambiente experimental para esses vizinhos. Portanto o aumento do número de vizinhos principalmente para a técnica baseada na correlação de Pearson, não afeta de nenhuma forma o desempenho dela.

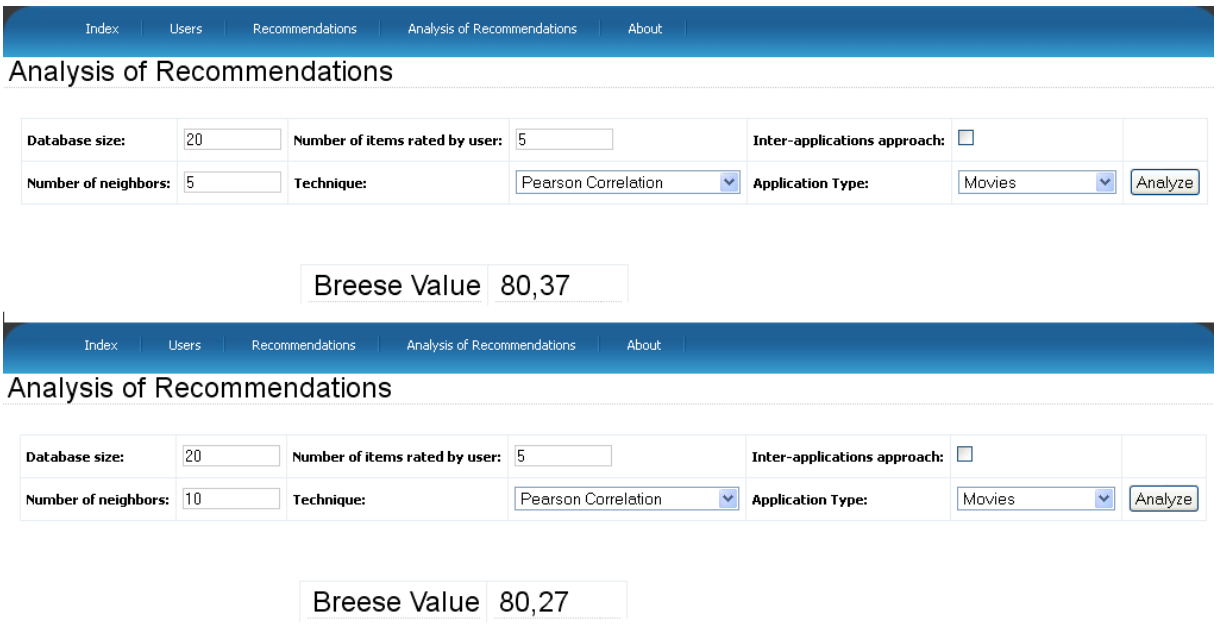

Figura 13. Resultados da métrica Breese utilizando o algoritmo de correlação de Pearson para o número de vizinhos igual a 5 e 10.

O algoritmo de filtragem baseado em perfis simbólicos apresentou um padrão similar ao da técnica baseada na correlação de Pearson no que diz respeito a número de vizinhos, em determinadas configurações de cenários. A mudança na quantidade de vizinhos mais próximos não alterou significativamente o desempenho desta técnica. Este algoritmo também apresentou nas mudanças de cenários dentro de um mesmo tamanho de base, uma variação mínima de desempenho, em algumas situações houve alteração para mais, em algumas outras os valores foram diminuídos.

A tabela ainda mostra que para os algoritmos baseados no coeficiente de correlação de Pearson e na distância euclidiana, a medida em que os cenários dentro de um mesmo tamanho de base passam a possuir um número maior de itens no perfil dos usuários, estas técnicas melhoraram seu desempenho em cada mudança feita.

Avaliando agora cada tamanho de base, na base de dados pequena, o algoritmo de filtragem baseado em perfis simbólicos consegui um melhor resultado em $\boldsymbol{R_5^{ps}}$ na configuração do cenário 1 e $\boldsymbol{R_{10}^{ps}}$ para o cenário 2. Ainda na base de dados pequena, no cenários 3, o algoritmo de filtragem baseado no coeficiente de correlação de Pearson teve o melhor desempenho para $\boldsymbol{R_5^{ccp}}$.

Para a base de dados de tamanho médio, no cenário 4 e cenário 5, as variáveis $\boldsymbol{R_{10}^{ps}}$ relacionadas ao algoritmo baseado em perfis simbólicos, apresentaram um melhor resultado. O cenário 6 teve o algoritmo baseado na distância euclidiana como o de melhor desempenho. A configuração representada pela variável $\boldsymbol{R_{10}^{de}}$ foi a que apresentou o melhor valor para a métrica Breese.

Já nos experimentos que utilizaram a base de dados de tamanho grande, os cenários 7, 8 e 9 tiveram os algoritmos de filtragem baseados em perfis simólicos, no coeficiente de correlação de Pearson e na distância euclidiana, como os de melhores resultados respectivamente. A variável $\boldsymbol{R_5^{ps}}$ para o cenário 7, $\boldsymbol{R_{10}^{ccp}}$ para o cenário 8 e $\boldsymbol{R_{10}^{de}}$ para o cenário 9, apresentaram os melhores resultados.

Mediante aos desempenhos apresentados em um sistema de recomendação tradicional, pode-se observar a capacidade do algoritmo baseado em perfis simbólicos em recomendar itens utilizando poucas informações sobre o usuário. Este algoritmo consegue sugerir itens desta forma, pois ele trabalha com a informação social dos usuários para encontrar os vizinhos mais próximos e calcular as similaridades.

Quando o número de itens no perfil do usuário começa a aumentar, os algortimos clássicos tem seu desempenho aumentado da mesma maneira. Estes algoritmos precisam conhecer o usuário em questão, pois esses algoritmos encontram vizinhos utilizando um modo item a item de procura.

4.2.2 Análise de técnicas de filtragem colaborativa utilizando abordagem inter-aplicações

Este experimento se propõem a avaliar as técnicas de filtragem colaborativa em um ambiente inter-aplicações. Como no experimento anterior, os cenários montados para a avaliação são os referentes à **Tabela 4** e a numeração citada nas análises também podem ser observadas nesta tabela.

A medição de qualidade das recomendações foi feita usando os três contextos de aplicações: filmes, livros e bandas. No momento da criação do perfil do usuário todos os itens que este usuário avaliou nos três contextos são levados em consideração. Desta forma o perfil que é montado pode ser composto por itens mistos, ou até mesmo, por itens de somente uma das três aplicações. Está é uma das diferença presente nas recomendações dos itens em relação a metodologia tradicional. A **Tabela 7**, mostra os resultados da métrica Breese para todo conjunto de itens em um ambiente inter-aplicações.

Tabela 7. Resultados referentes aos valores da métrica de Breese para um sistema de recomendação utilizando a abordagem inter-aplicações.

	Pequena			***Média***			***Grande***		
	8	*5*	*2*	*8*	*5*	*2*	*8*	*5*	*2*
R_5^{ps}	79,21	82,63	80,54	82,17	83,73	**81,2**	82,3	81,33	**83,61**
R_{10}^{ps}	81,95	**83,29**	**83,42**	79,48	**84,11**	80,23	83,55	82,77	80,86
R_5^{de}	82,6	81,41	75,56	83,11	78,88	74,82	83,57	82,44	75,73
R_{10}^{de}	82,44	83,12	78,81	**83,3**	82,29	76,51	83,96	82,73	77,64
R_5^{ccp}	**82,87**	80,73	72,24	83,02	80,75	73,93	85,25	84,25	74,26
R_{10}^{ccp}	82,74	80,08	72,5	82,56	79,26	73,71	**85,78**	**84,76**	74,87

Nota-se que os resultados da métrica Breese em um ambiente inter-aplicações também foram altos. Os valores apresentados seguem o nível de desempenho semelhante ao do experimento anterior, mas com alguns padrões diferentes. Durante as simulações, todos os cenários propostos foram sujeitos à criação de um perfil do usuário sem ao menos um item da própria aplicação que utilizou o serviço de integração inter-aplicações. Isto significa que em muitos casos o

sistema a princípio não sabia praticamente nada a respeito do usuário, mas conseguiu recomendar itens satisfatoriamente.

O comportamento das técnicas com relação a sensibilidade na mudança do número de vizinhos, foi praticamente o mesmo ao encontrado nos experimentos que utilizaram a abordagem tradicional. Nos resultados do algoritmo de Pearson é possível notar tal situação facilmente.

Outro ponto a ser observado é que esta abordagem tratou bem o problema do novo usuário, apesar de definirmos em todos os cenários o número máximo de itens avaliados. Nos cenários em que o perfil do usuário é composto de apenas 2 itens e 5 itens, como mostra a Figura 14, esta abordagem apresentou resultados interessantes para valores relacionados à base de dados pequena e grande, respectivamente.

Pequena			***Média***			***Grande***		
8	*5*	*2*	*8*	*5*	*2*	*8*	*5*	*2*
79,21	82,63	80,54	82,17	83,73	**81,2**	82,3	81,33	**83,61**
81,95	**83,29**	**83,42**	79,48	**84,11**	80,23	83,55	82,77	80,86
82,6	81,41	75,56	83,11	78,88	74,82	83,57	82,44	75,73
82,44	83,12	78,81	**83,3**	82,29	76,51	83,96	82,73	77,64
82,87	80,73	72,24	83,02	80,75	73,93	85,25	84,25	74,26
82,74	80,08	72,5	82,56	79,26	73,71	**85,78**	**84,76**	74,87

Figura 14. Pontos em que os resultados da abordagem inter-aplicações foram visivelmente melhor.

Percebe-se também que a variação do desempenho de um cenário que apresenta número de itens no perfil igual a 5, para 8, diminuiu em relação ao experimento anterior. Os resultados obtidos nos cenários em que envolvem esses perfis, se manteram altos e também mais próximos.

Avaliando os cenários divididos pelo tamanho das bases de dados, começaremos com a base pequena. O algoritmo baseado em perfis simbólicos teve melhor desempenho para a configuração do cenário 1 na variável $\boldsymbol{R_{10}^{ps}}$. Para o cenário 2 este mesmo algoritmo teve a média da métrica Breese acima dos demais

para a variável $\boldsymbol{R_{10}^{ps}}$. E no cenário 3, o algoritmo baseado no coeficiente de correlação de Pearson teve melhor desempenho para a configuração referente a variável $\boldsymbol{R_{5}^{ccp}}$.

Para a base de dados média, o algoritmo de melhor resultado para o perfil de usuário de tamanho igual a 2 e 5 foi novamente o algoritmo baseado em perfis simbólicos. As variáveis que apresentaram este rendimento foram as $\boldsymbol{R_{5}^{ps}}$ e $\boldsymbol{R_{10}^{ps}}$, respectivamente. No cenário 6, o algoritmo baseado na distância euclidiana foi o que teve melhor resultado na variável $\boldsymbol{R_{10}^{de}}$.

E por fim a avaliação para base de dados grande. O cenário 7 apresentou através da variável $\boldsymbol{R_{5}^{ps}}$ o melhor desempenho, associado ao algoritmo baseado em perfis simbólicos. Os cenário 8 e 9 tiveram o mesmo algoritmo como o de melhores resultados. As configurações para esses cenários foram feitas através da variável $\boldsymbol{R_{10}^{ccp}}$, para o algoritmo de filtragem baseado no coeficiente de correlação de Pearson.

Com base nesses resultados, verifica-se a capacidade elevada de um sistema de recomendação que utiliza uma abordagem inter-aplicações em sugerir itens. O problema do novo usuário e o da ovelha negra foi praticamente eliminado diante dos resultados apresentados. Esta abordagem trabalha com perfis de usuários de várias aplicações para poder melhorar a capacidade de recomendação do sistema.

Outro ponto que pode ser destacado, é que o bom nível de qualidade das recomendações se deve também a capacidade dos algoritmos que usaram esta abordagem em encontrar vizinhos mais parecidos com o usuário em questão, pois itens em outros contextos também foram observados e comparados.

Capítulo 5

Conclusões e Trabalhos Futuros

Neste último capítulo, estão apresentadas todas as contribuições realizadas a partir deste trabalho. Também estão listadas as conclusões finais sobre os resultados da análise proposta e, por fim, são apresentadas algumas idéias para trabalhos futuros.

5.1 Contribuições

Este trabalho forneceu um conjunto de análises das técnicas de filtragem colaborativa em um sistema de recomendação que suporta recomendações tradicionais e inter-aplicações. Uma ferremanta de análise foi desenvolvida para suportar os experimentos deste trabalho, além de servir de base para futuras implementações de sistemas de recomendação nesse estilo.

Os resultados encontrados no capítulo anterior a partir dos experimentos propostos, mostraram que em um ambiente de recomendação tradicional a técnica de filtragem colaborativa que utilizou o algoritmo baseado nos perfis simbólicos, obteve um melhor desempenho no cenário com baixo número de usuários e poucos itens avaliados por eles. Já nos cenários onde o perfil de usuário possuía 8 itens, o algoritmo baseado na distância euclidiana apresentou um desempenho interessante para todas as base de dados propostas. O algoritmo baseado no coeficiente de correlação de Pearson apresentou, em geral, valores próximos aos obtidos pelas técnicas de melhores resultados.

Os resultados referentes a abordagem inter-aplicações se mostraram promissores. Para as configurações dos cenários onde o perfil do usuário possuía número de itens igual a 2 ou 5, esta abordagem demonstrou ser capaz de sugerir itens com um nível de qualidade satisfatório. Este fato é muito interessante, pois em aplicações reais o usuário não costuma fornecer uma grande quantidade de informações em relação a suas preferências. Deste modo o usuário poderá se

agradar com as recomendações geradas pelo sistema de recomendação inter-aplicações mesmo sem ter usado muito este sistema.

Quanto aos resultados referentes aos perfis de usuários para um número de 8 itens avaliados, restringindo a formação do perfil do usuário a uma única aplicação, obversamos que a qualidade das recomendações geradas pelo sistema mantém-se num mesmo patamar para os algoritmos avaliados segundo a métrica Breese. Com isso podemos dizer que a abordagem inter-aplicações para sistemas de recomendação traria vários benefícios para aplicações que utilizassem essa abordagem para sugerir listas personalizadas de itens aos seus usuários. Além de ser uma abordagem inovadora, ela tem a capacidade de apresentar recomendações surpreendentes, uma vez que fogem à regra de itens conhecidos pelo usuário em questão [4].

5.2 Dificuldades e Desafios

Dentre as dificuldades que podem ser levantadas, a construção da base de dados é uma delas. Atualmente são utilizadas em experimentos que envolvem sistemas de recomendação, bases de dados disponíveis na internet. No nosso caso não foi possível utilizá-las, pois as avaliações presentes nessas bases se restrigiam a uma única aplicação. Assim foi necessária a coleta de preferências e modelagem da base de dados para o desenvolvimento de um sistema de recomendação inter-aplicações. Outro fator que pode ser citado envolve também a base de dados. Para uma avaliação profunda do desempenho dos algoritmos, o número de preferências dos usuários nas aplicações usadas em ambiente inter-aplicações devem ser bem maior, ponto este levantado na seção de trabalhos futuros.

A construção de um sistema de recomendação com uma abordagem tão inovadora foi empolgante pela forma como o sistema foi pensado e modelado. A primeira etapa no desenvolvimento foi implementar os três algortimos de filtragem colaborativa na sua forma tradicional e logo depois adicionar a funcionalidade inter-aplicações a todos eles. Feito isso, foi modelada a arquitetura do sistema e os pontos de integrações entre as aplicações, seguindo padrões arquiteturais e padrões de projeto. Por fim houve a integração das funcionalidades com a parte de análise, para assim colher todos os resultados apresentados no capítulo anterior.

5.3 Trabalhos Futuros

O desenvolvimento de todo este trabalho trouxe também algumas idéias para a construção de novos trabalhos sobre sistemas de recomendação usando a abordagem tradicional e a abordagem inter-aplicações, dentre as quais são listadas abaixo:

- Utilizar uma base de dados maior para a análise de qualidade das recomendações geradas pelo sistema de recomendação.
- Aumentar o número de contextos envolvidos nas recomendações inter-aplicações.
- Usar outras métricas de medição de qualidade, como Spearsman[9] e Kendall Tau[9], e confrontá-las dentro do ambiente de recomendações.
- Propor novos métodos de filtragem de informação, adaptados para sistemas de recomendação inter-aplicações, buscando explorar melhor as potencialidades que este tipo de sistema provê.

Bibliografia

[1] Zhi-Dan Zhao, Ming-sheng Shang, **User-Based Collaborative Filtering Recommendation Algorithms on Hadoop,** *Proceeding WKDD '10 Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining IEEE Computer Society Washington*, DC, USA ©2010

[2] Taek-Hun Kim, Young-Suk Ryu, Seok-In Park, and Sung-Bong Yang, **An Improved Recommendation Algorithm in Collaborative Filtering**, *Dept. of Computer Science, Yonsei University, Seoul*, 120-749, Korea, 2002

[3] Xiaoyuan Su and Taghi M. Khoshgoftaar, **A Survey of Collaborative Filtering Techniques**, *Department of Computer Science and Engineering, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA Received 9 February 2009; Accepted 3 August 2009*

[4] De Amorim, J. S., **Um Serviço de Recomendação Inter-aplicações Baseado em Filtragem Colaborativa**. 2010. Trabalho de Conclusão de Curso (Engenharia da Computação) – Escola Politécnica – Universidade de Pernambuco, Recife-PE

[5] Bezerra, B. L. D. **Soluções em personalização de conteúdo baseadas em classificadores simbólicos modais**, Universidade Federal de Pernambuco. Recife, p. 211. 2008.

[6] BIRUKOU, A. et al. **IC-service - A service-oriented approach to the development of recommendation systems**. *Proceedings of the* 2007 *ACM symposium on Applied computing*. Seoul: ACM. 2007. p. 1683 -1688

[7] D. Billsus and M. Pazzani, **Learning collaborative information filters**, *in Proceedings of the 15th International Conference on Machine Learning (ICML '98)*, 1998.

[8] Ralph, P. Parkson, J. A Framework for Automatic Online Personalization. Proceedings of the 39th Annual Hawaii International Conference on System Sciences. 2006. p. 137b – 137b.

[9] Jonathan L. Herlocker, Joseph A. Konstan , Loren G. Terveen , John , T. Riedl, **Evaluating Collaborative Filtering Recommender Systems,** ACM T*ransactions on Information Systems*, 2004

[10] Alag, Satnam, **Collective Intelligence in Action**, *Greenwich, CT:* Manning Publications Co, (2009)

[11] Schwab, I., Kobsa, A. *and* Koychev I., **Learning User Interests through Positive Examples Using Content Analysis and Collaborative Filtering,** Internal Memo, GMD, 2001

[12] R. Burke, **Hybrid recommender systems: survey and experiments**. User Modeling and User-Adapted Interaction, Department of Information Systems and Decision Sciences, California State University, Fullerton, 2002.

[13] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, **Analysis of recommendation algorithms for E-commerce**, in *Proceedings of the ACM E-Commerce*, pp. 158–167, Minneapolis, Minn, USA, 2000

[14] BELL, R. M.; KOREN, Y. **Improved neighborhood-based collaborative filtering**. KDD Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2007.

[15] P. Melville, R. J. Mooney, and R. Nagarajan, **Content-boosted collaborative filtering for improved recommendations**, in *Proceedings of the 18th on National Conference on Artificial Intelligence (AAAI '02)*, pp. 187–192, Edmonton, Canada, 2002.

[16] Bezerra B. L. D., De Carvalho F. A. T., Macário W., **Um Método de Filtragem Híbrida Baseado em Perfis Simbólicos Colaborativos**, Centro de Informática – Universidade Federal de Pernambuco, Recife, PE, Brasil

[17] G. Adomavicius and A. Tuzhilin, **Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions**, *IEEE Transactions on Knowledge and Data Engineering* **17** (2005), 634–749.

[18] SHARDANAND, U.; MAES, P. **Social information filterin - Algorithms for automating "word of mouth"**. Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems. Denver: ACM Press/Addison – Wesley Publishing Co. 1995. p. 210-217.

[19] Alves, V. S. **Um Algoritmo Evolutivo Rápido para Agrupamento de Dados.** 2007. Dissertação (Mestrado em Ciência da Computação) Curso de Pós-graduação em Ciência da Computação, Universidade Católica de Santos, São Paulo

[20] Redpath Jennifer, Shapcott M., McClean S. and Chen L., ***A Study of Evaluation Metrics for Recommender Algorithms***. In: *The 19th Irish Conference on Artificial Intelligence and Cognitive Science* (2008).

[21] Herlocker, Jonathan Lee. **Understanding and Improving Automated Collaborative Filtering Systems.** 2000, Thesis (Doctor of Philosophy), University of Minnesota, USA.

[22] Breese JS, Heckerman D, Kadie C (1998) **Empirical analysis of predictive algorithms for collaborative filtering**. In: Cooper GF, Moral S (eds) *Proceedings of the 14th conference on uncertainty in artificial intelligence* (UAI-98). Morgan Kaufmann, San Francisco, pp 43–52

[23] Endrei, Mark; Ang, Jenny; Arsanjani , Ali; Chua, Sook; Comte, Philippe; Krogdahl, PŒl; Luo, Min; Newling, Tony, **Patterns: Service-Oriented Architecture and Web Services,** IBM Redbooks, 2004

[24] Sharp, John, **Microsoft Windows Communication Foundation 4 – Step by Step,** California: O'Reilly Media, Inc, 2010.

[25] ERL, T. **SOA Design Patterns**. 1st. ed. Indiana: Pearson Education, Inc, 2008.