UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM

INSTITUTO DE COMPUTAÇÃO - ICOMP

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

ANDERSON PIMENTEL DOS SANTOS

SISTEMA DE RECOMENDAÇÃO BASEADO EM AGRUPAMENTO USANDO PROPAGAÇÃO DE AFINIDADES

Manaus, AM

2017

ANDERSON PIMENTEL DOS SANTOS

SISTEMA DE RECOMENDAÇÃO BASEADO EM AGRUPAMENTO USANDO PROPAGAÇÃO DE AFINIDADES

Projeto Final apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas - UFAM como requisito parcial para a obtenção do grau de Mestre em Informática.

Orientador: Prof. D.Sc. Edleno Silva de Moura

Manaus, AM

2017

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Santos, Anderson Pimentel dos
S237s Sistema de Recomendação Baseado em Agrupamento usando Propagação de Afinidades / Anderson Pimentel dos Santos. 2017
44 f.: il.; 31 cm.

Orientador: Edleno Silva de Moura
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. sistema de recomendação. 2. filtragem colaborativa. 3. agrupamento. 4. propagação de afinidades. I. Moura, Edleno Silva de II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"SISTEMA DE RECOMENDAÇÃO BASEADO EM AGRUPAMENTO USANDO PROPAGAÇÃO DE AFINIDADES"

ANDERSON PIMENTEL DOS SANTOS

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Edleno Silva de Moura - PRESIDENTE

Prof. César Augusto Viana Melo - MEMBRO INTERNO

Prof. João Marcos Bastos Cavalcanti - MEMBRO EXTERNO

Manaus, 01 de Setembro de 2017

Resumo

Recomendar itens baseados na similaridade de interesses (Filtragem Colaborativa) é atrativo para muitos domínios: livros, filmes, músicas, produtos e etc, mas isso nem sempre funciona bem devido ao fato das coleções de itens serem muitos esparsas, como ocorre em empresas como Amazon, Netflix, Spotify, entre outras.

A Filtragem Colaborativa baseada em agrupamento propõe maior escalabilidade para coleções muito esparsas, sua premissa é que se a pessoa a e pessoa b gostam de um mesmo conjunto de filmes, então provavelmente a pessoa a gostará de outros filmes que a pessoa b gosta. Aglomerando pessoas em grupos baseados nos itens que elas compraram, pode-se obter boas recomendações de itens a serem comprados, dessa forma, as predições podem ser feitas aglomerando-se pessoas em grupos em função dos filmes que elas assistem(agrupamento baseado no usuário) e/ou grupos de filmes que tendem a ser do gosto das mesmas pessoas (agrupamento baseado no item).

O K-means é um algoritmo clássico de agrupamento, sendo simples, eficiente e amplamente utilizado, entretanto podemos citar algumas limitações em seu uso, como, o número de grupos que deve ser definido a priori, a sensibilidade à escolha inicial dos centróides na criação dos grupos, a possibilidade de gerar grupos vazios, entre outros.

O algoritmo Propagação de Afinidades é uma alternativa ao k-means, é um algoritmo proposto recentemente que ganhou grande popularidade na aplicação em áreas da bioinformática, apresentando bons resultados para problemas de agrupamentos de sequências de DNA, mas também vem sendo aplicado em outras áreas, como agrupamento de faces (imagem), coleções de filmes e na sumarização de textos.

Neste trabalho é apresentada a implementação do algoritmo Propagação de Afi-

nidades em sistemas de recomendação baseados em agrupamento, com o intuito de investigar se os bons resultados que o algoritmo tem mostrado em outras áreas são válidos também para a área de recomendação de vídeos baseada em agrupamento, realizando comparações entre coleções de filmes por meio de métricas de avaliação de predição para sistemas de recomendação.

Palavras-chave: sistema de recomendação, filtragem colaborativa, agrupamento.

Abstract

Recommend items based on similarity of interests (Collaborative Filtering) is attractive to many domains: books, movies, music, products and etc. However, it's not always works well due to the fact of collections of items as scattered as in companies such as Amazon, Netflix, Spotify, among others.

Clustering based collaborative filtering proposes greater scalability for very sparse collections, its premise is if the person a and person b like the same set of movies, then the person a probably will like other movies the person b likes. Clustering people into groups based on the items they bought, one can get good recommendations for items to be bought, in this way, predictions can be made by crowding people into groups, based on the movies they watched(user-based) and/or groups of movies which tend to be of the taste of the people the same interest(item-based).

The K-means is a classic clustering algorithm, being simple, efficient and widely used, however, it comes with some restrictions: the number of final groups must be defined *a priori*, very sensitive to the initial choice of centroids in the creation of groups, it can generate empty groups, among others.

The algorithm Affinity Propagation is an alternative to K-means, it is a recently proposed algorithm that has gained great popularity in areas of bioinformatics, presenting good results for problems like clustering DNA sequences, and being applied also for clustering of faces (image), collections of films and summarization of texts.

The document presents an approach of Recommender Systems based on clustering using Affinity Propagation in order to investigate whether the good results that the algorithm has in other areas are also valid for recommender systems area.

Keywords: recommender system, collaborative filtering, clustering.

"Aquele que dá bons conselhos, constrói com uma mão. Aquele que dá bons conselhos e exemplo, constrói com ambas". Dedico este trabalho aos meus pais, meus guias presentes em todos momentos de minha vida.

Agradecimentos

- Primeiramente a Deus por ser essencial em minha vida, autor do meu destino e meu guia.

- Ao amigo e orientador, Prof. Dr. Edleno de Moura pela sua grande experiência e conhecimento na direção e orientação deste trabalho, além de seu grande desprendimento em ajudar em momentos difíceis vividos durante esse período de mestrado. É uma imensa honra tê-lo como orientador.

- Agradeço também a todos os funcionários e professores do IComp, que me acompanharam durante o mestrado, em especial ao Prof. Dr. Altigran da Silva, Prof. Dr. Marco Cristo, Prof. Dr. Eulanda Santos, Prof. Dr. David Fernandes, Prof. Dr. Moisés Carvalho, Prof. Dr. César Melo, Prof. Dr. Arilo Dias Neto, Prof. Dr. João Cavalcanti e Prof. Dr. Eduardo Souto por compartilharem um pouco de seus conhecimentos.

- Aos meus irmãos Marla Laís e Lucas Anghinoni, que sempre me apoiaram incondicionalmente, que acreditaram em mim e que me passaram a segurança e certeza de que não estou sozinho nessa caminhada.

- À Laiza Serrão, ouvinte atenta de algumas dúvidas, inquietações, desânimos e sucessos. Agradeço de coração pelo apoio, confiança e pela valorização entusiasmada do meu trabalho, dando-me desta forma, coragem para ultrapassar a culpa pelo tempo que a cada dia lhe subtraía.

- À CAPES pelo auxílio financeiro para o desenvolvimento deste trabalho.

- A todos aqueles que contribuíram direta ou indiretamente para a realização deste trabalho, seja com incentivo nas tarefas de pesquisa ou horas de descontração. Em especial aos amigos da Pós-Graduação: Airton, Alternei, Awdren, Bernardo, Bruno, Euler, Fernando, Hugo, Kayro, Luis, Marcelo, Omar, Ralph, Renato, Richard, Roland, Tiago e Tony. Aos amigos do Teewa e Akiry: Antonio, Caio, Daniel, Berg, Jackson, Maiara e Raphael, e aos amigos do laboratório GTI(BDRI): Anibrata, Diego Barros, Diego Rodrigues, Leonardo, Ludimila, Caio, Joyce, Josiane, Ivanilse, Márcio Palheta, Denys Silveira e Luisa.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

Sistemas de Recomendação (SR) foram criados como ferramentas para auxiliar na descoberta de novas informações de interesse de um determinado usuário. Para isso, utilizam de técnicas que proveem sugestões personalizadas de itens para uso de um usuário (Ricci et al., 2010), selecionando e apresentando opções personalizadas que possam ser de seu interesse, melhorando sua navegação e experiência (Ribeiro et al., 2014).

Os SR podem auxiliar usuários a encontrar seus itens de interesses utilizando predições baseados em seu comportamento passado (Kuzelewska, 2014). Um típico sistema de recomendação gera como saída uma lista de recomendação para atender aos interesses do usuário, as principais aplicações em SR se baseiam nas técnicas de: filtragem baseada em conteúdo, filtragem demográfica, filtragem colaborativa (FC) e filtragem híbrida. A filtragem baseada em conteúdo utiliza conteúdo de perfis e/ou descrição de produtos associados a usuários e itens para a recomendação. Na abordagem de FC, a recomendação é feita para um determinado usuário com base nos padrões de preferência de outros usuários por meio das avaliações atribuídas aos itens. A FC é bem popular por ser simples, prática e de fácil implementação (Liao and Lee, 2016), pois se baseia no fato de que um usuário pode ter interesse sobre itens que foram bem avaliados por usuários similares a ele (Ricci et al., 2010), além de não necessitar de domínio de conhecimento, que dificilmente pode ser obtido e também pode ser um indicativo de má qualidade. A filtragem demográfica é justificada no princípio de que usuários

com atributos pessoais (sexo, idade, país e etc.) compartilham preferências. E por fim, a filtragem híbrida, que combina o uso de FC com filtragem demográfica, ou ainda, com filtragem baseada em conteúdo, onde normalmente utiliza métodos bioinspirados ou probabilísticos, tais como algoritmos genéticos, redes neurais, redes bayesianas e algoritmos de agrupamento (Bobadilla et al., 2013).

Recomendar itens baseados na similaridade de interesses (FC) é atrativo para muitos domínios: livros, filmes, músicas, produtos e etc, mas isso nem sempre funciona bem devido ao fato dos dados serem esparsos (Ungar and P. Foster, 2000). Em coleções de dados de empresas como Amazon[1], Netflix[2] ou Spotify[3], a quantidade de dados é gigantesca, acompanhada de sua esparsidade.

Classificadores de FC utilizam muitas operações que envolvem cálculo de distância, por exemplo, para achar os *k-nearest neighbors*[4] (*KNN*). Contudo, para coleções com uma enorme quantidade de itens, esses métodos não são escaláveis, e mesmo com diminuição da dimensionalidade ainda teriam que tratar de várias distâncias entre objetos.

Nesse contexto, podemos considerar o uso de algoritmos de agrupamento, que podem ajudar na eficiência porque reduzem o número de operações de distância, apesar de poder comprometer a acurácia (Ricci et al., 2010).

Agrupamento é referenciado na literatura como um método não superviosionado de aprendizagem e consiste em atribuir itens para grupos de forma que itens de um mesmo grupo são mais similares quando comparado com itens de outros grupos (Ricci et al., 2010), sendo o objetivo a descoberta natural de grupos (que tenham algum significado) em uma coleção. A similaridade dos itens é determinada usando funções de distância (Rongfei et al., 2010)(Bobadilla et al., 2013). Os objetivos dos algoritmos de agrupamento são minimizar a distância intra-grupos e maximizar a distância inter-grupos (Ricci et al., 2010).

A premissa para FC baseada em agrupamento é que se a pessoa a e pessoa b gostam

[1] http://www.amazon.com
[2] http://www.netflix.com
[3] http://www.spotify.com
[4] k vizinhos mais próximos

de um mesmo conjunto de filmes, então provavelmente a pessoa *a* gostará de outros filmes que a pessoa *b* gosta (Kuzelewska, 2014). Aglomerando pessoas em grupos baseados nos itens que elas compraram, pode-se obter boas recomendações de itens a serem comprados, dessa forma, as predições podem ser feitas aglomerando-se pessoas em grupos em função dos filmes que elas assistem (agrupamento baseado no usuário) e/ou grupos de filmes que tendem a ser do gosto das mesmas pessoas (agrupamento baseado no item) (Ungar and P. Foster, 2000)(Rongfei et al., 2010)(Wei et al., 2012).

O *k-means* é um algoritmo clássico de agrupamento, sendo extremamente simples, eficiente e amplamente utilizado em tarefas de agrupamento (Ricci et al., 2010). O algoritmo trabalha com centróides que são normalmente escolhidas aleatoriamente, onde os itens são atribuídos aos grupos mais próximos. Os centróides são atualizados continuamente até não haver mais mudança de itens entre grupos (dado um limiar de distância) (Ungar and P. Foster, 2000).

Existem algumas limitações conhecidas sobre o *k-means* original. Podemos citar alguns: (1) conhecimento a *priori* dos dados, para a escolha do número k de grupos que será gerado; (2) os grupos gerados no final são muito sensíveis à escolha inicial dos centróides; (3) pode produzir grupos vazios (Ricci et al., 2010).

Uma alternativa ao *k-means* é o algoritmo de agrupamento *Propagação de Afinidades*[5] (PA). É um algoritmo proposto recentemente que ganhou grande popularidade na aplicação em áreas da bioinformática, apresentando bons resultados para problemas de agrupamentos de sequências de DNA, mas também vem sendo aplicado em outras áreas, como agrupamento de faces (imagem) (Bodenhofer et al., 2011), combinado a outros métodos em coleções de filmes (Amatriain, 2013), e na sumarização de textos (Ricci et al., 2010).

O PA é um algoritmo que utiliza a troca de mensagens para criar grupos que ao invés de considerar um conjunto inicial de pontos como centros e iterativamente adaptá-los, o algoritmo por troca de mensagens inicialmente considera todos os pontos como centros, chamados de exemplares (Dueck and Frey, 2007). Durante a execução

[5]do inglês *Affinity Propagation*

dos pontos no algoritmo, que são considerados como nós em uma rede de nós, acontecem as trocas de mensagens até que os grupos gradualmente apareçam, assim, o número de agrupamentos é gerado automaticamente, sem precisar de um parâmetro *a priori*, como acontece no *k-means*. O algoritmo PA é um representante importante dessa família de algoritmos e define dois tipos de mensagem: (1) responsabilidade, que representa quão bem (acúmulo de evidências) o recebedor da mensagem serve para ser um exemplar, levando em conta outros potenciais exemplares; (2) disponibilidade, que é enviada do candidato a exemplar para o recebedor da mensagem, e reflete quão apropriado(acúmulo de evidências) é o recebedor para escolher o candidato a exemplar como seu exemplar (Frey and Dueck, 2007).

A proposta deste trabalho é implementar o algoritmo PA em sistemas de recomendação baseados em agrupamento, com o intuito de investigar se os bons resultados que o algoritmo tem mostrado são válidos também para a área de recomendação de vídeos baseada em agrupamento, realizando comparações entre coleções de filmes por meio de métricas de avaliação de predição para sistemas de recomendação.

Neste trabalho é apresentada a aplicação de PA em sistemas de recomendação, propondo duas contribuições. A primeira é a apresentação de três técnicas de recomendação utilizando *k-means* como algoritmo principal do sistema de recomendação, onde os resultados são obtidos com base em quatro coleções de filmes e por meio de duas métricas de avaliação de resultados. Na segunda contribuição, o PA é aplicado como algoritmo principal na recomendação de filmes, observando a variação do fator de amortecimento para obtenção de resultados com diferente tempo de término do algoritmo de agruapamento e número de grupos gerado.

Este texto está organizado da seguinte forma: O próximo capítulo apresenta conceitos essenciais que servem como base para o entendimento do trabalho, onde aborda trabalhos significativos na literatura relacionados com a proposta deste documento. No capítulo 3 é descrita a implementação do algoritmo de agrupamento PA em um sistema de recomendação baseado em agrupamento. No capítulo 4 são mostrados os resultados dos experimentos, e por fim, no capítulo 5, as conclusões sobre a investigação do uso

do algoritmo de agrupamento PA em sistemas de recomendação.

Capítulo 2

Conceitos e Trabalhos Relacionados

Neste capítulo serão apresentados conceitos sobre sistemas de recomendação, algoritmos de agrupamento e trabalhos relacionados. Inicialmente, introduzimos o problema principal de recomendação e um resumo sobre os principais métodos citados na literatura. Logo após, serão apresentados alguns métodos de agrupamento e as funções de similaridades que são utilizadas neste trabalho.

2.1 Sistemas de Recomendação

2.1.1 Conceito

Os Sistemas de Recomendação (SR) foram criados como ferramentas para auxiliar na descoberta de novas informações de interesse de um determinado usuário. Para isso, utilizam de técnicas que proveem sugestões personalizadas de itens para uso de um usuário (Ricci et al., 2010), selecionando e apresentando opções personalizadas que possam ser de seu interesse, melhorando sua navegação e experiência (Ribeiro et al., 2014). Os SR utilizam o termo geral *item* para referir-se ao que recomendam para o usuário, normalmente são baseados em um tipo de item específico, como filmes, livros, serviços, músicas, entre outros (Ricci et al., 2010). Também será muito usado o termo *usuário ativo*, referenciando o usuário a quem o sistema gera as recomendações.

O processo de geração de recomendação em um SR frequentemente é baseado na

combinação dos seguintes aspectos (Bobadilla et al., 2013):

- **Tipo de dados disponíveis**: avaliações, informações de registro de usuários, características e conteúdo de itens que podem ser ranqueados, informações de relacionamentos sociais entre usuários e informações de localização;
- **Algoritmo de filtragem**: Filtragem demográfica, baseada em conteúdo, colaborativa, sensível ao contexto e híbrida;
- **Modelo escolhido**: baseado no uso direto dos dados (*memory-based*[1]) ou modelo usando os dados(*model-based*[2]);
- **Emprego de Técnicas**: Abordagens probabilísticas e bioinspiradas como redes *bayesianas*, algoritmos de vizinhos mais próximos, redes neurais, algoritmos genéticos, algoritmos de agrupamento;
- **Nível de Esparsidade**: Esparsidade das coleções e escalabilidade desejada;
- **Objetivo**: Predições ou lista *top*-n de recomendações;
- **Qualidade dos Resultados**: Novidade, diversidade, métricas de precisão, revocação, *RMSE*[3], *MAE*[4], entre outras.

A seguir apresenta-se uma discussão a respeito de cada um desses aspectos.

Tipos de Dados Disponíveis

Pesquisas em SR utilizam coleções públicas de dados que sejam significativas em seus contextos, de forma que facilite a investigação das técnicas, métodos e algoritmos desenvolvidos por pesquisadores da área. Por meio dessas coleções, a comunidade científica pode replicar experimentos para validar e/ou melhorar suas técnicas (Bobadilla et al., 2013). O grupo de pesquisa *GroupLens Research*[5] disponibiliza várias coleções com

[1]Abordagem baseada em memória
[2]Abordagem baseada em modelo
[3]Root-mean-square error
[4]Mean Absolute Error
[5]https://www.grouplens.org

representatividade na área de pesquisa de recomendação de filmes, algumas de suas coleções são utilizadas neste trabalho e serão apresentadas no próximo capítulo.

2.1.2 Algoritmos de Filtragem

Funções internas para SR são caracterizadas como algoritmos de filtragem. Normalmente são dividos em algoritmos (Ricci et al., 2010)(Kuzelewska, 2014): (1) filtragem demográfica, (2) filtragem baseada em conteúdo, (3) filtragem colaborativa e (4) filtragem híbrida.

Filtragem Demográfica

Justificada no príncipio de que os indivíduos com certos atributos pessoais (sexo, idade, país, idioma e etc.) em comum, também compartilham de preferências comuns (Bobadilla et al., 2013). Muitos *websites* adotam essa abordagem. Por exemplo, o sistema pode identicar a localização de onde o usuário está acessando o *website*, país,idioma, idade. Dessa forma, sugerindo itens personalizados de acordo com esses atributos (Ricci et al., 2010).

Filtragem Baseada em Conteúdo

Nessa técnica, os sistemas aprendem a recomendar itens que são similares aos que o usuário gostou no passado (Ricci et al., 2010). A similaridade é calculada baseada nas características do item a ser comparado. Na verdade, seria fácil recomendar um novo livro de *O Senhor dos Anéis* para um usuário, se nós soubermos que esse livro se trata de um livro de fantasia e que o usuário gosta dele. Um recomendador pode realizar essa tarefa se baseando na descrição do novo livro de *O Senhor dos Anéis* e nas descrições de interesse do usuário (Lacerda and Ziviani, 2013).

2.1.3 Filtragem Colaborativa

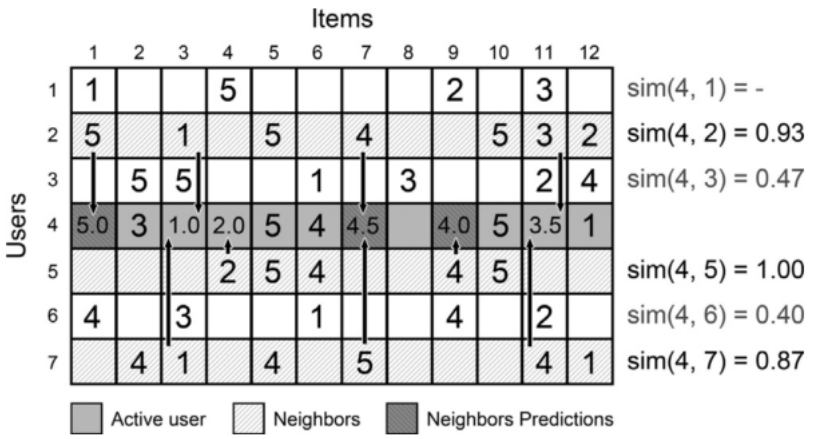
Nessa técnica utiliza-se o conhecimento adquirido com um conjunto de usuários para realizar recomendações. Por exemplo, ao recomendar itens populares para um usuário,

o sistema usa informação colaborativa, obtida a partir do padrão de comportamento de outros usuários, para selecionar tais itens. Também pode ser considerada colaborativa qualquer recomendação que tome como base a similaridade de um usuário com uma dada comunidade de usuários. Quando a recomendação colaborativa utiliza informação passada do usuário combinada ao comportamento da comunidade, diz-se que a mesma é híbrida, por combinar aspectos de conteúdo com aspectos colaborativos.

Abordagem Baseada em Memória

A abordagem baseada em memória produz a recomendação colaborativa de itens por meio da manipulação direta da base de dados durante a seleção de itens a serem recomendados. Ela tenta encontrar em tempo real usuários que são similares ao usuário para quem se deseja gerar recomendações e usa as preferências de usuários similares (figura 2.1) para tentar predizer as preferências do usuário que receberá recomendações.

Figura 2.1: Ilustração do Mecanismo de Predição baseado em média utilizando *kNN*



Fonte: (Bobadilla et al., 2013)

Abordagem Baseada em Modelo

Sistemas de recomendação baseados em memória podem apresentar a desvantagem de ter custo alto e eventual lentidão no processo de recomendação, podendo portanto gerar problemas de escalabilidade, especialmente quando deseja-se gerar recomendação de itens em tempo real sobre grandes bases de dados. Uma solução para tais problemas de escalabilidade é o uso de uma abordagem baseada em modelos.

Sistemas de recomendação baseados em modelos envolvem a construção de um modelo de recomendação obtido por meio de um processamento prévio da base de dados do sistema. Em outras palavras, extrai-se informação da base de dados para gerar um modelo e apenas esse modelo é usado durante a seleção de itens a serem recomendados. O benefício potencial dessa estratégia é a possibilidade de desenvolver-se modelos que são ao mesmo tempo extremamente rápidos e de baixo custo computacional, aumentando assim as chances de se desenvolver um sistema altamente escalável.

Abordagem Baseada em Agrupamento

A abordagem baseada em agrupamento pode ser classificada como uma abordagem baseada em modelo, dado que o agrupamento prévio de usuários permite a geração de recomendações com base na informação de agrupamento ao invés do uso direto da base de dados.

Filtragem Híbrida

Esses SR são baseados na combinação das técnicas de filtragem. A filtragem híbrida combina técnicas a e b, de modo que tenta usar as vantagens de a para resolver problemas da b. Por exemplo, métodos de FC têm a desvantagem de usar novos itens adicionados na coleção, pois não podem recomendar itens que ainda não tem avaliações do usuário (Ricci et al., 2010). Entretanto, isso não é um problema para métodos de filtragem baseada em conteúdo, desde que a predição de novos itens seja baseada em suas descrições, que normalmente é facilmente disponível. Ou seja, dado duas ou mais técnicas básicas de SR, combinadas das mais variadas formas para criar novos sistemas híbridos de recomendação (Kuzelewska, 2014)(Li and Kim, 2003).

2.1.4 Objetivo da Recomendação

Quando considera-se o objetivo do sistema, pode-se dividi-los em dois tipos básicos: os de predição de resultados e os de lista de resultados. Os sistemas de predição de resultados funcionam como classificadores que têm como objetivo classificar os itens

da base de dados em adequados para serem recomendados ou não adequados. Neles não há a noção de ordem de resultados. Nos sistemas de listas de resultados, o sistema não tem a missão de classificar os itens a serem recomendados, limitando-se a produzir uma lista ordenada de candidatos a serem boas recomendações para um usuário. A lista é ordenada por alguma estimativa de relevância, funcionando de maneira similar a um sistema de busca.

2.1.5 Qualidade dos Resultados

A qualidade esperada dos resultados também é um dos aspectos importantes a serem considerados em um sistema de recomendação. Um aspecto importante quando se fala em qualidade é a apresentação de resultados ao mesmo tempo diversos, novos e relevantes. Em geral, a cada recomendação que pode ser feita para um usuário, há a possibilidade de se enviar diversos tipos distintos de itens. Quando se fala em diversidade, espera-se que o sistema seja capaz de trazer representantes de cada tipo possível, permitindo assim que o usuário tenha uma ampla visão do que existe na base e que possa ser de seu interesse. A inclusão de novidades também é um aspecto importante, pois por meio dela o usuário pode ter contato com novos tipos de itens descobrir novos interesses dentro da base de dados. Finalmente, a combinação dos aspectos de novidade, popularidade e relevância é uma tarefa importante para que se produza um bom sistema de recomendação. Os três eixos devem estar equilibrados para maximizar a utilidade do recomendador.

2.2 Recomendação de Filmes

O domínio de entretenimento, mais especificamente a recomendação de filmes, é um domínio bastante explorado na literatura de sistemas de recomendação. Isso se deve ao fato de que as pesquisas em sistemas de recomendação, além da contribuição teórica, visam geralmente melhorias na indústria de sistemas de recomendação e envolvem pesquisas sobre diversos aspectos práticos que podem ser aplicados na implementação

de sistemas de recomendação (Ricci et al., 2010).

A *Netflix* anunciou o prêmio *Netflix Prize*, em 2006. Um evento importante para a pesquisa, comunidade e indústria de sistemas de recomendação. Isso mostrou a importância que as recomendações de items para os usuários pode acelerar o desenvolvimento de novas técnicas e abordagens de mineração de dados para recomendação. Apesar do *Netflix Prize* iniciar uma grande quantidade de atividades de pesquisa, o prêmio foi a simplificação de um problema real de recomendação. Esse problema consiste na predição de uma nota de usuário para um filme de forma que o *RMSE* (métrica de avaliação de qualidade da predição) seja otimizado na comparação entre a nota predita e a nota real do usuário (Ricci et al., 2010).

Figura 2.2: Ilustração de Avaliação de Título de Filme pelo Usuário

Fonte: *http://www.techtudo.com.br/*

Um sistema de recomendação de filmes espera notas dos usuários para os filmes assistidos, como na figura 2.2. Para recomendação por filtragem colaborativa, o uso das avaliações dos usuários para os filmes pode ser representado como uma matriz Usuário-Item ou matriz U-I (Shi et al., 2014), como na tabela 2.1.

Tabela 2.1: Tabela Matriz Usuário-Item

	Titanic	Inception	Toy Story	Taken	Skyfall	Matrix
Ricardo	5	?	3	?	?	1
Erica	?	1	?	4	?	?
Jean	2	4	?	?	?	5
Roberto	?	2	?	?	3	?

2.3 Métodos de Agrupamento

O agrupamento é uma tarefa de aprendizagem não supervisionado para organização ou particionamento de dados em grupos significativos (Dueck and Frey, 2007). Serão apresentados nessa seção as técnicas de agrupamento *K-means* e o algoritmo de agrupamento *Progação de Afinidades*.

2.3.1 K-means

K-means é uma técnica clássica de agrupamento, onde K representa o número de grupos (que deve ser especificado) em que o algoritmo deve subdividir os dados. Após definir o número de grupos, o algoritmo escolhe aleatoriamente os representantes daqueles grupos, os chamados centróides de grupo. Logo em seguida, todas as instâncias de dados são atribuídas a sua centróide mais próxima, de acordo com a métrica de distância euclidiana. Depois é feito um cálculo para definir novos valores para os centróides de cada grupo, fase chamada de *means*. Esse processo é repetido até que as centróides estabilizem, ou seja, permaneçam as mesmas.

Essa técnica de agrupamento é simples, eficaz e fácil de provar que ao repetir as iterações para encontrar a centróide, é possível minimizar a distância total ao quadrado entre os todos pontos dos grupos e suas centróides (mínimo local). Porém, não é possível garantir que este resultado é o mínimo global, visto que qualquer mudança na escolha aleatória inicial dos representantes pode acarretar em resultados totalmente diferentes, causando uma sensibilidade dos resultados finais em relação a escolha dos centros iniciais. Porém, uma alternativa para encontrar o mínimo global é executar o algoritmo várias vezes com combinações de escolhas diferentes, e escolher aquele com o melhor resultado.

O algoritmo do k-means é dividido em três fases, que são:

(1) Calcular valores iniciais para k grupos

Nessa fase inicial, como mostrado no algoritmo 1, o algoritmo escolhe k instâncias de dados de forma aleatória, chamadas de centróides, que irão representar cada um dos k

grupos, para em seguida começar as iterações para encontrar o melhor resultado.

(2) Atribuir objeto a centróide mais próxima

Nessa etapa, é calculado para todas as instancias de dados, a distância euclidiana em relação a todas as centróides criadas na fase anterior. Assim, cada instância fará parte do grupo pertencente à centróide mais próxima.

(3) Calcular média da distância entre os pontos

Depois de todos os grupos criados com seus objetos, a primeira iteração do algoritmo é calcular a distância média entre todos os pontos pertencentes a cada grupo, e então, determinar este ponto como a nova centróide. Devido à mudança de centróide, a distância entre todos os pontos em relação às novas centróides são recalculados, por isso alguns pontos podem mudar de grupo. Essa fase de calcular a média da distância entre os pontos ocorre repetidas vezes, até que não ocorram mudanças nas centróides, que então estarão estabilizadas.

Algoritmo 1: Algoritmo K-means

```
calcula valores iniciais para k grupos m[k] (1)
atribui objeto ao centróide mais próximo (2)
while existir mudanças na posição das médias (3) do
  estima as médias para classificar nos grupos
  for i=1 to k do
    m[i] ← média das amostras para o grupo i
  end for
end while
```

2.3.2 Propagação de Afinidades

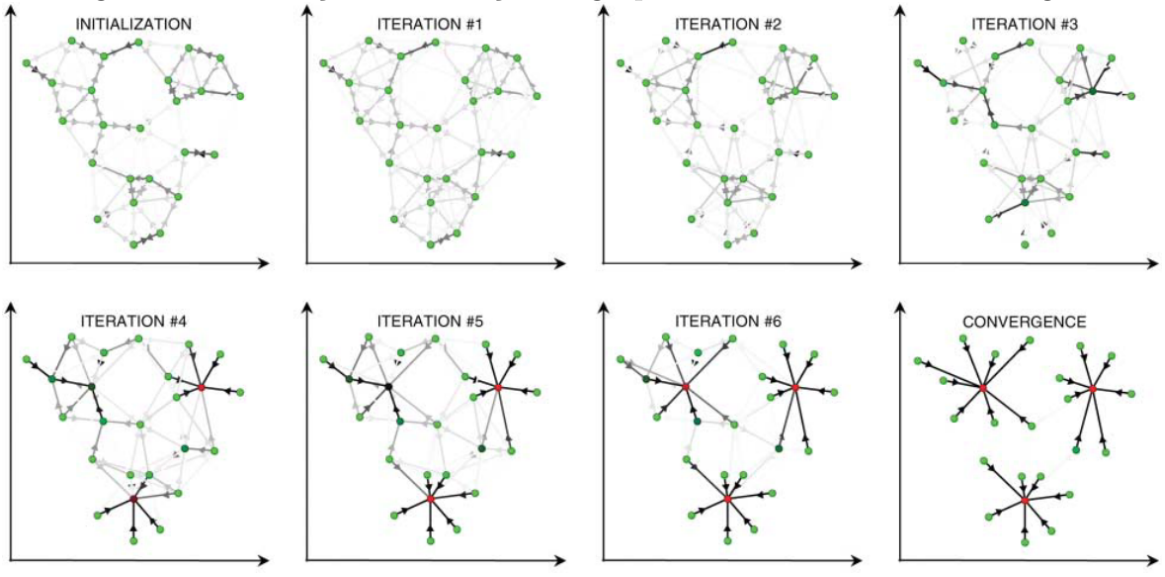
Algoritmo que identifica exemplares entre um conjunto de itens e gera grupos de itens ao redor desses exemplares. Isto ocorre considerando simultaneamente todos os itens como possíveis exemplares e ocorrendo a troca mensagens entre os itens até emergir conjuntos de exemplares e seus respectivos grupos (Bodenhofer et al., 2011). O termo *data point* será chamado de item, se refere aos objetos que serão agrupados, já o termo

"exemplar" se refere ao item que é representante de si e de outros itens. Portanto, o algoritmo de Propagação de Afinidades (PA) cria grupos de itens em torno de cada exemplar que emergiu durante as trocas de mensagens entre itens.

Agrupamento de dados baseados em medidas de similaridade são uma etapa crítica na análise científica dos dados e engenharia de sistemas (Frey and Dueck, 2008). Uma abordagem comum é usar dados para aprender a agrupar um conjunto de centros, de forma que a soma dos erros quadráticos entre dois itens e seus centros seja menor. Quando os centros são selecionados de pelo item atual, eles são chamados de "exemplares" (Frey and Dueck, 2007).

Esse algoritmo tem uma abordagem diferente e considera simultaneamente todos os itens como possíveis exemplares. Vendo cada item como um nó de uma rede, o método recursivamente transmite mensagens valores através de arestas da rede até que um conjunto de exemplares e correspondentes grupos apareçam. Ou seja, os nós da rede de nós iniciam com potencial para ser um exemplar, sendo que a cada iteração são realizadas trocas de evidências ou mensagens, chamadas de *responsabilidade* e *disponibilidade*. A própria rede de nós identifica, através de uma quantidade definida de iterações, quais os nós são mais apropriados para ser exemplares da rede, por meio das trocas de mensagens, como ilustrado na figura 2.3.

Figura 2.3: Ilustração de formação de grupos durante a troca de mensagens



Fonte: (Frey and Dueck, 2008)

É importante observar que o fator de amortecimento do algoritmo de propagação de afinidades pode cooperar para a convergência mais rápida do algoritmo, apesar de poder influenciar na quantidade final de grupos.

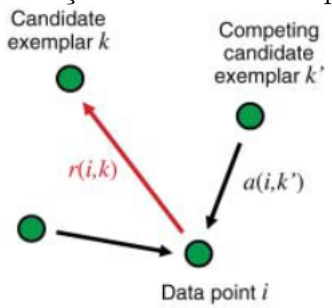
O algoritmo PA espera como entrada uma coleção de similaridades entre os itens, onde $s(i, k)$ indica quão bem o item de índice k é adequado como exemplar do item i. Ao invés de necessitar de uma entrada definindo a quantidade de grupos que serão gerados, o PA toma como entrada números reais de similaridade $s(k, k)$ para cada item k, de forma que o item com valor alto de $s(k, k)$ têm maior probabilidades de ser exemplar, estes valores são referidos como *preferências* (Frey and Dueck, 2007).

A coleção de similaridades de entrada pode ser obtida no domínio de recomendação de filmes utilizando a matriz U-I, como apresentado na seção 2.2. As similaridades entre entre os usuários ou itens podem ser calculadas utilizando as funções de similaridades que serão apresentadas na seção 2.4. Dessa forma, podemos criar uma matriz de similaridade usuário-usuário ou item-item, dependendo do tipo de abordagem que será empregada.

O número de exemplares identificados (número de grupos) é bastante influenciado pelos valores da entrada preferências, mas também podem aparecer pela troca de mensagens entre itens. A priori, se todos os itens são igualmente adequados como exemplares, as preferências devem ser ajustadas para um valor comum ou podem ser alteradas para encontrar diferentes quantidades de grupos. Um valor que pode ser o valor comum para um conjunto de itens é a mediana das similaridades desse conjunto, o que resulta em um número moderado de grupos, podemos assumir o menor valor das similaridades (resultando em menor número de grupos) ou então o maior valor (resultando em maior número de grupos)(Frey and Dueck, 2007).

Com relação as mensagens trocadas entre os itens, existem duas, e cada uma leva em consideração um diferente tipo de competição. As mensagens podem ser combinadas em qualquer estágio para decidir que itens são exemplares, e para os outros itens, qual exemplar pertence a quem. A *responsabilidade* $r(i, k)$ enviada de um item i para um candidato a exemplar k (figura 2.4), reflete o acúmulo de evidências de quão bem

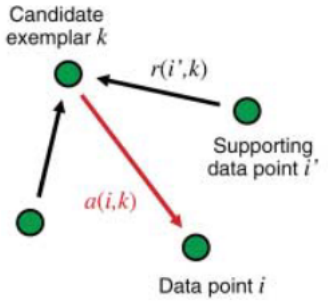
Figura 2.4: Ilustração do envio de responsabilidades



Fonte: (Frey and Dueck, 2008)

adequado o item k é para servir como exemplar do item i, levando em consideração os outros potenciais exemplares do item i (Frey and Dueck, 2008). A *disponibilidade* $a(i,k)$ enviada do item candidato a exemplar k para o item i (figura 2.5), reflete o acúmulo de evidências para quão apropriado seria se o item i escolhesse o item k como seu exemplar, levando em consideração o suporte de outros itens que o item k deve ser exemplar (Frey and Dueck, 2008).

Figura 2.5: Ilustração do envio de disponibilidade



Fonte: (Frey and Dueck, 2008)

O algoritmo inicia com a disponibilidade dos itens igual a 0: $a(i,k) = 0$ e as responsabilidades são computadas usando a regra definida na equação 2.1.

$$r(i,k) \leftarrow s(i,k) - \max_{j:j\neq k}(a(j,i) + s(i,j)) \tag{2.1}$$

Na primeira iteração, como as disponibilidades são zero, o $r(i,k)$ é inicialmente a similaridade de entrada entre os itens i e k como seu exemplar, menos a maior similaridade entre o item i e os outros candidatos a exemplar. Esta atualização competitiva é baseada nos dados e não leva em consideração como quais itens favorece para cada

candidato a exemplar (Dueck and Frey, 2007). Nas últimas iterações, quando alguns itens são efetivamente atribuídos a outros exemplares, suas disponibilidades irão cair para abaixo de zero, como descrito na equação 2.2.

$$a(k,i) \leftarrow \min(0, r(k,k) + \sum_{j:j\notin\{k,i\}} \max\{0, r(j,k)\} \tag{2.2}$$

As disponibilidades negativas diminuem efetivamente os valores de similaridade de entrada $s(i,k')$, removendo o correspondente candidato a exemplar da competição de ser um exemplar (Dueck and Frey, 2007).

Para $k = i$, a responsabilidade $r(k,k)$ é atribuida a preferência de entrada para o item k ser escolhido como exemplar, $s(k,k)$, menos a maior similaridade entre os itens i e todos os outros candidatos a exemplar. Isso é chamada de *autoresponsabilidade* e reflete o acúmulo de evidências que o item k é um exemplar, baseado em sua preferência de entrada combinada com quão não adequado é para ser atribuído para outros exemplares.

Para a disponibilidade $a(i,k)$ é atribuida a autoresponsabilidade somada as responsabilidades positivas que o candidato a exemplar k recebe dos outros itens. Apenas porções positivas de responsabilidades de entrada são adicionadas. Isso serve como evidência para o item, de forma que indica que representar bem outros itens (responsabilidades positivas) ou que pode fracamente representar outros itens (responsabilidades negativas).

Para limitar a forte influência da chegada de responsabilidades positivas, o total do somatório apresentado na equação 2.3 é limitado de forma que não pode receber valores negativos.

$$a(k,k) \leftarrow \sum_{j:j\neq\{k,i\}} \max\{0, r(j,k)\} \tag{2.3}$$

A mensagem de autodisponibilidade evidencia que o item k é um exemplar, baseado na responsabilidade positiva enviada do candidato a exemplar k para os outros itens.

Na atualização das mensagens é comum utilizar um procedimento de amortecimento

para evitar oscilações numéricas que podem aparecer em algumas circunstâncias. Cada mensagem é atribuída λ vezes a seu valor da iteração passada, somada com $1 - \lambda$ vezes seu valor previamente atualizado (Frey and Dueck, 2007), onde o fator de amortecimento λ varia entre 0 e 1, contudo, experimentos revelam um fator padrão para evitar oscilações númericas no valor de 0.9.

O algoritmo PA foi proposto em recentemente e ganhou grande popularidade na aplicação em áreas da bioinformática, apresentando bons resultados para problemas de agrupamentos de sequências de DNA, mas também vem sendo aplicado em outras áreas, como agrupamento de faces (imagem)(Bodenhofer et al., 2011), combinado a outros métodos em coleções de filmes (Amatriain, 2013), e na sumarização de textos (Ricci et al., 2010). No trabalho proposto em Tacchini and Damiani (2011), foi utilizado o PA em um sistema de recomendação de músicas utilizando a coleção *Last.fm*, usando como entrada a matriz músicos-usuários para agrupar os músicos, e propondo medidas de similaridade que apresentaram grupos com significado, seja comparado a origem dos cantores ou estilo musical.

2.4 Funções de Similaridade

As métricas de similaridade determinam a similaridade entre pares de usuários (em FC usuário-usuário), ou similaridade entre itens (em FC item-item). Para este contexto, podemos comparar avaliações de todos os itens avaliados por dois usuários, ou as avaliações que todos os usuários deram para dois itens (Ricci et al., 2010).

Métodos de classificação e técnicas de agrupamento são normalmente utilizados em abordagens de FC, esses métodos e técnicas dependem altamente da definição apropriada de similaridades ou métricas de distância (Rongfei et al., 2010).

2.4.1 Distância Euclidiana

O mais simples e comum exemplo de métrica de distância é a distância Euclidiana:

$$d(x, y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2}, \tag{2.4}$$

onde n é o número de dimensões (atributos) e x_k e y_k são os k^{th} atributos (componentes) dos objetos x e y, respectivamente (Ricci et al., 2010).

Para o cálculo da similaridade com base na distância euclidiana:

$$s(x, y) = \frac{1}{1 + d(x, y)}. \tag{2.5}$$

2.4.2 Cosseno

A similaridade entre dois documentos pode ser medida tratando cada documento como um vetor de frequência de palavras, e calculando o cosseno do ângulo entre os dois vetores de frequência (Su and Khoshgoftaar, 2009). Esta função pode ser usada em FC, onde podemos usar usuários e itens ao invés de documentos, e avaliações no lugar das frequências de palavras.

Teoricamente, se R é uma matriz m x n usuário-item, então a similaridade entre dois objetos (usuários ou itens), i e j, é definida como o cosseno do ângulo entre vetores de dimensão n.

$$s(\vec{x}, \vec{y}) = \frac{\vec{x} \bullet \vec{y}}{\|\vec{x}\| \times \|\vec{y}\|}. \tag{2.6}$$

Onde "$\bullet$" significa o produto escalar entre dois vetores.

2.4.3 Correlação

A similaridade entre objetos pode ser dada por usa *correlação* que mede a relação linear entre os objetos. Existem muitas *correlações* que podem ser aplicadas, mas a mais utilizada é a correlação de *Pearson*. Dada a covariância dos pontos x e y, $\sum$ e

seu desvio padrão σ, podemos calcular a correlação de *Pearson* usando:

$$s(x, y) = \frac{\sum(x, y)}{\sigma_x \times \sigma_y}. \tag{2.7}$$

Neste capítulo foram introduzidos os conceitos principais de sistemas de recomendação, sendo apresentada na seção 2.1.1 as considerações que devem ser levadas em conta na criação de um sistema de recomendação. Foram também apresentadas técnicas de agrupamento *K-means* e Propagação de Afinidades na seção 2.3, e por fim as funções de distância e similaridades na seção 2.4.

Capítulo 3

Implementação do Sistema

Este trabalho tem a proposta de investigar o impacto de uso do algoritmo de agrupamento *Propagação de Afinidades* (PA) na qualidade das recomendações por meio das predições de nota de filmes em Sistemas de Recomendação (SR). A implementação deste sistema de recomendação foi baseado no algoritmo de recomendação proposto em Kuzelewska (2014). O trabalho é realizado utilizando diferentes coleções públicas de filmes.

As imagens e algoritmos dessa seção são baseados no agrupamento de usuários, entretanto, suas aplicações podem ser ajustadas para grupos baseados em itens com o mesmo objetivo de recomendar itens de interesse para o usuário, os resultados dos experimentos apresentados no capítulo 4 foram realizados utilizando as duas abordagens.

A implementação do sistema de recomendação utiliza o método de Filtragem Colaborativa (FC) com a abordagem baseada em agrupamento para fazer predições de avaliações de filmes em coleções de filmes. O algoritmo de recomendação proposto pode ser dividido em duas etapas:

1. **Offline**: Agrupamento de objetos similares e construção de um modelo de dados contendo os perfis dos usuários/itens;

2. **Online**: Geração de recomendação dos filmes (predição de avaliações).

A lista de recomendações e os procedimentos gerais são apresentados no algoritmo 2, onde a entrada do algoritmo depende da matriz de avaliações de itens realizadas pelos

usuários, com a quantidade de usuários n, a quantidade de itens (filmes) k e c avaliações realizadas pelos usuários para os itens. A similaridade entre os usuários/itens pode ser obtida utilizando-se a função de similaridade δ.

Algoritmo 2: Algoritmo Geral do Sistema de Recomendação

Entrada:

- M = (U,I,A) − matriz u-i, onde $U = \{u_1, \ldots, u_n\}$ é o conjunto de usuários, $I = \{i_1, \ldots, i_k\}$ é o conjunto de itens e $A = \{a_1, \ldots, a_c\}$ é o conjunto de avaliações dos itens pelo usuário;
- δ − a função de similaridade;
- fa $\in [0.5, 0.9]$ − fator de amortecimento do algoritmo Propagação de Afinidades;
- pref − preferências do algoritmo Propagação de Afinidades.

Resultado:

- $G = \{G_1, \ldots, G_nc\}$− conjunto de grupos;
- $G_r = \{g_{r,1}, \ldots, g_{r,ng}\}$− conjunto de representantes de grupo;
- R_{u_a}− lista de itens recomendados para um usuário ativo.

$G \leftarrow agruparDados(M, A, \delta, fa, pref)$;
$G_r \leftarrow calcularRepresentantes(M, A, G)$;
$R_{u_a} \leftarrow recomendar(u_a, G_r)$;

Os grupos são criados utilizando-se o método de agrupamento PA e sua entrada é uma matriz de similaridade entre objetos (usuários ou itens). Para criar os ng grupos utilizando PA são necessários os seguintes parâmetros: fator de amortecimento fa que visa resolver problemas de oscilações nos resultados (pode aumentar a velocidade da convergência de resultados), e as preferências *pref*, que por padrão utiliza a mediana dos valores de similaridade para apresentar os pontos de amostra mais adequados a exemplares.

O método de agrupamento PA é usado para criar grupos de usuários/itens. De posse dos grupos, é então realizada a combinação das avaliações contidas na matriz u-i (matriz usuário-item), criando-se assim os perfis para usuários/itens dos grupos. Nessa etapa (*offline*), ou seja, que é realizada em um momento anterior ao da recomendação de itens, os custos do sistema podem ser mais altos, dado que o tempo de processamento desta

etapa não afeta o tempo de recomendação percebido pelos usuários. Na segunda etapa, são geradas as recomendações baseadas unicamente nos modelos de representantes de grupos criados na etapa *offline*.

Após a geração dos grupos de objetos é necessária a criação de um modelo, ou perfil, que descreve cada um dos grupos. Vamos identificar tal modelo como sendo o modelo de *representantes dos grupos*. Um representante de grupo é um vetor de notas/avaliações que representa os interesses do grupo, contendo a combinação de notas dos filmes do grupo que o modelo representa. Na etapa de recomendação, o modelo é consultado e a predição de itens a serem recomendados ocorre baseada nas notas desse vetor. Dado um usuário que precisa receber recomendações, identifica-se inicialmente a que grupo o usuário pertence. A informação sobre o grupo é então utilizada como informação de entrada para a seleção de itens a serem recomendados com base modelo representante do grupo.

Estudamos duas formas alternativas propostas na literatura para a criação dos representantes de grupos: A primeira é a Técnica de Geração de Representantes de Grupos baseada em Frequência de Valores de Avaliação, apresentada na seção 3.1 e a segunda é a Técnica de Geração de Representantes de Grupos baseada em Frequência de Maiores e Menores Valores de Avaliação, apresentada na seção 3.2.

Nossa decisão para estudar técnicas baseadas em agrupamentos para realizar recomendação se deu pelo fato dessa abordagem ter resultado final bastante efetivo e, ao mesmo tempo, ser uma técnica bastante escalável e propícia para o uso em sistemas de recomendação *online*. Como as preferências de um usuário ativo são comparadas apenas com os representantes dos grupos, e a estimativa de notas é realizada usando apenas valores do vetor de representantes do grupo, a complexidade final da fase *online* é $O(k \cdot ng)$, ou seja, depende apenas do número de grupos ng e do número de itens k.

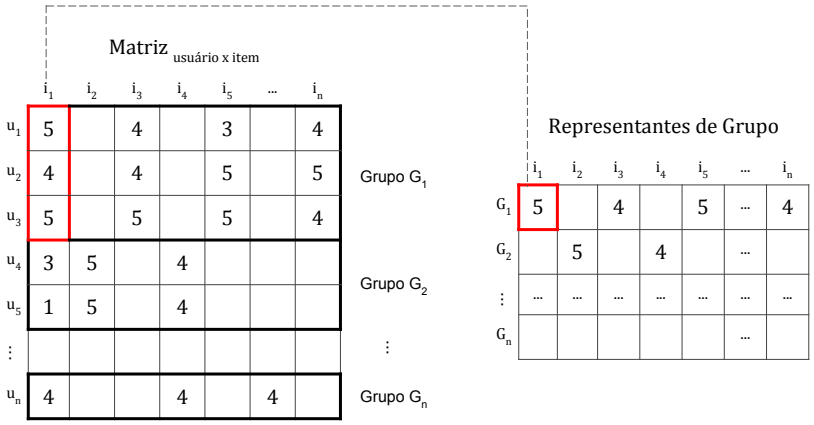
3.1 Técnica de Geração de Representantes de Grupos baseada em Frequência de Avaliações

Essa técnica de geração de representantes de grupos visa encontrar as notas/avaliações de filmes mais frequentes em um grupo de objetos. Como descrito na equação 3.1, quando o número de notas mais frequentes for maior que 1, o valor do representante final será igual a 0.

$$A_p(G_q) = \begin{cases} a_l, & \forall a_l \in A \qquad a_l \overset{max|u(i_p)|}{=} u(i_q); \\ 0, & \text{para os outros casos.} \end{cases} \tag{3.1}$$

A figura 3.1 apresenta um exemplo com o uso da técnica de criação de representantes de grupo utilizando as frequências de notas. Como no grupo G_1, para o filme i_1, a nota mais frequente é a nota 5, para o vetor de representantes do grupo G_1, o filme i_1 terá 5 como nota representante, de maneira análoga para esse grupo com o filme i_3 a nota representante será 4.

Figura 3.1: Ilustração do mecanismo de predição usando representantes de grupos baseados em Frequência de Valores de Avaliação



3.2 Técnica de Geração de Representantes de Grupos baseada em Frequência de Maiores e Menores Valores de Avaliação

Na equação 3.2, a frequência de p maiores ou menores são examinadas. Se o número de notas p maiores é igual ou superior a $\alpha \cdot n$, a nota final será a média das p maiores notas. Se, predominantemente, os valores são p menores são mais frequentes, então a nota final será a média das p menores notas.

A ideia dessa técnica é que se no conjunto de notas contiver notas muito extremas (por exemplo, 10 notas 5, uma nota 1 e uma nota 2), se apresentando como um ruído na técnica de utilizar as frequências das notas, essa notas ruidosas são eliminadas e consideradas apenas as notas mais frequentes em um limiar definido de frequência de notas (maiores ou menores).

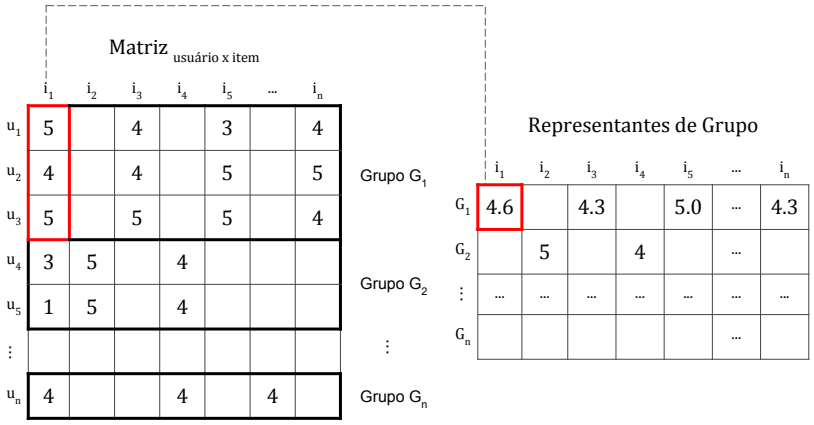
$$A_p(C_q) = \begin{cases} \overline{u(i_p)}, & \forall \in \{c-p \dots c\} \quad |u(i_p)| \geq \alpha \cdot n; \\ \overline{u(i_p)}, & \forall \in \{1 \dots p\} \quad |u(i_p)| \geq \alpha \cdot n; \\ 0, & \text{para outros casos.} \end{cases} \tag{3.2}$$

Para definir o limiar em um contexto onde as notas dos filmes variam entre 1 a c, sendo $c = 5$, uma boa prática seria utilizar o valor mediano para definir as duas partições. Portanto, o valor de 3 é um ótimo valor em um intervalo $[1; 5]$, separando as possibilidades de nota em dois subintervalos: (1) o intervalo de 1 a 2, que será considerado como o de valores menores; (2) o intervalo 4 a 5, que será considerado como sendo o de valores maiores. A nota 3 é excluída da contagem. A idéia é que uma nota intermediária representa uma informação neutra e não dá pistas claras sobre o interesse ou desinteresse do usuário pelo filme.

Podemos observar na figura 3.2 o cálculo de notas para obtenção dos vetores representates do grupo. Nessa técnica, as frequências são contabilizadas para os intervalos de maiores ou menores valores.

Na figura 3.2, o grupo G_1 tem 3 notas no intervalo de maiores valores para o filme

Figura 3.2: Ilustração do mecanismo de predição usando representantes de grupos baseados em frequências de maiores e menores valores



i_1, portanto, de acordo com a equação 3.2, a nota no vetor de representantes será a média das 3 notas, que resultou em *4.6*. Um outro exemplo, no mesmo grupo G_1, agora com relação ao filme i_5, contamos duas notas 5 e uma nota 3, só que a nota 3 não está em nenhum dos dois intervalos que contabilizam para frequências dos intervalos de maiores e menores notas. Então, calculadas as duas notas 5, a média gera a mesma nota 5.

Capítulo 4

Experimentos

4.1 Configuração dos Experimentos

4.1.1 Coleções Utilizadas

Neste trabalho foram utilizadas coleções de títulos de filmes para a recomendação de títulos para os usuários do sistema. Cada coleção contém avaliações explícitas feitas pelos usuários na escala de 1 a 5. Detalhes sobre a quantidade de usuários, avaliações e títulos de filmes de cada coleção são apresentados abaixo:

LDOS-CoMoDa

LDOS-CoMoDa (Odic et al., 2013) é uma coleção para recomendação de filmes baseada em contexto. Possui avaliações explícitas dos filmes e 12 partes de informações relacionadas ao contexto que descrevem em que situação os filmes foram assistidos.

A coleção LDOS-CoMoDa conta com 2296 avaliações de 1232 títulos de filmes realizadas por 121 usuários. Um ponto importante dessa coleção é que os dados adquiridos na avaliação ocorrem imediatamente após os usuários assistirem os filmes. Cada usuário submete uma avaliação para o filme que assistiu e também adiciona em sua submissão o contexto de sua experiência, como por exemplo: o horário do dia, o tipo de dia (dia da semana, fim de semana ou feriado), estação, localização, tempo, companhia enquanto assistia (social), emoções dominantes durante e ao término do filme, humor, condições

físicas, descoberta(seleção própria ou sugerido por outros) e interação.

MovieLens

As coleções MovieLens (Harper and Konstan, 2015) são disponibilizadas pelo grupo de pesquisa GroupLens (Research, 2017) do departamento de Ciência da computação e Engenharia da Universidade de Minnesota. As coleções foram armazenadas em vários períodos de tempos, dependendo do tamanho da coleção. As coleções utilizadas em nossos experimentos foram:

MovieLens ML2K No Segundo *Workshop* em Heterogeneidade de Informação e Fusão em Sistemas de Recomendação (HetRec) de 2011 foram lançadas algumas coleções muito citadas em trabalhos relacionados a sistemas de recomendação. Entre essas coleções estava a MovieLens 200K que contém *tagging*, informações de localização e informações sobre as avaliações explícitas dos usuários. Possui 2113 usuários, com 855.598 avaliações sobre os 10197 títulos de filmes.

MovieLens ML100K A MovieLens ML100K consiste em 943 usuários, com 100.000 avaliações, sendo que cada usuário tem pelo menos 20 avaliações, em um total de 1682 títulos de filmes.

MovieLens 1M A coleção MovieLens 1M contém arquivos com 1.000.209 avaliações anônimas de 6.040 usuários para aproximadamente 3.900 títulos de filmes. O mínimo de 20 avaliações por usuário.

A tabela 4.1 apresenta os dados de esparsidade das coleções. A esparsidade mede o quão esparsas são as matrizes que relacionam usuários a filmes dentro de cada coleção. Quanto maior o índice de esparsidade, maior a quantidade de entradas nulas dessa matriz.

Tabela 4.1: Tabela de Esparsidade das coleções

LDOS-CoMoDA	**ML-1M**	**ML-2K**	**ML-100K**
98,46%	95,53%	95,99%	93,7%

4.1.2 Métricas de Avaliação

Para avaliar o resultado dos sistemas implementados, adotamos métricas de avaliação de predição que são normalmente adotadas na literatura para comparar sistemas de recomendação. As duas métricas de avaliação mas frequentemente utilizadas na avaliação de sistemas de recomendação são a RMSE (*Root-mean-square Error*) e MAE (*Mean Absolute Error*).

RMSE

Conhecida como raiz do erro médio quadrático (*root mean squared error*), é a métrica mais popularmente utilizada em experimentos para avaliar acurácia na predição de avaliações de usuário. O sistema gera predições p_i para um conjunto de tamanho n, onde se sabe a avaliação correta q_i (Ricci et al., 2010).

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(p_i - q_i)^2}{n}} \tag{4.1}$$

MAE

Métrica alternativa e também bastante utilizada na literatura, MAE considera a média de magnitude dos erros em um conjunto de predições, sem considerar suas direções (Ricci et al., 2010).

$$MAE = \frac{\sum_{i=1}^{n}|p_i - q_i|}{n} \tag{4.2}$$

4.1.3 Algoritmo para Avaliação de Resultados

Para fins de reprodução, apresentamos a seguir o pseudocódigo dos algoritmos utilizados no cômputo das métricas de qualidade em nossos experimentos.

No algoritmo 3 são apresentados os passos para que seja calculada as métricas RMSE e MAE. A princípio, são selecionados testes aleatórios *nTeste* da matriz de notas usuário-filme. Após isso, o sistema realiza predições de 20% dos itens de cada usuário, calculando a diferença com as notas reais, e por fim, resultando no erro das

Algoritmo 3: Algoritmo de Avaliação dos Resultados Gerados pelo Sistema de Recomendação

Entrada:

- M = (U,I,A) − matriz u-i, onde $U = \{u_1, \ldots, u_n\}$ é o conjunto de usuários, $I = \{i_1, \ldots, i_k\}$ é o conjunto de itens e $A = \{a_1, \ldots, a_c\}$ é o conjunto de avaliações dos itens pelo usuário;
- $G_r = \{g_{r,1}, \ldots, g_{r,nc}\}$ − Conjunto de representantes dos grupos;
- nTeste − tamanho do conjunto de teste usado para validação;
- nItens − quantidade de itens para validação.

Resultado:

- RMSE − raiz do erro médio quadrático (*root mean squared error*)
- MAE − erro médio absoluto (*mean average error*)

$U_{teste} \leftarrow obterTesteAleatorio(U, nTeste)$;
for $u_p \in U_{teste}$ **do**
  ERROR $\leftarrow 0$;
  SQUARED_ERROR $\leftarrow 0$;
  itens $\leftarrow obterItensAleatorios(u_p, I)$;
  for $item_q \in nItens$ **do**
    $avaliacao_{est} \leftarrow estimarPref(p, q, U, G_r)$;
    ERROR $\leftarrow |(u_p(item_q) - avaliacao_{est}|$;
    SQUARED_ERROR $\leftarrow (u_p(item_q) - avaliacao_{est})^2$;
  end for
end for
MAE $\leftarrow ERROR/nTeste$
RMSE $\leftarrow SQUARED\_ERROR/nTeste$
RMSE $\leftarrow \sqrt{RMSE}$

predições.

No algoritmo 4 é apresentada a abordagem de recomendação de filmes utilizada neste trabalho. Os grupos de usuários ou itens são gerados utilizando a matriz de similaridades como entrada, como apresentado na seção 2.3.2. Para obter a predição de um determinado usuário p é necessário achar o grupo que o usuário/item está inserido e utilizar o representante do grupo (que já foram gerados a partir das técnicas de geração de representantes de grupo) G_r para verificar qual nota mais apropriada para aquele grupo, dado o filme q fornecido.

Algoritmo 4: Algoritmo de Predição de Notas de um Filme

Entrada:

- M = (U,I,A) − matriz u-i, onde $U = \{u_1, \ldots, u_n\}$ é o conjunto de usuários, $I = \{i_1, \ldots, i_k\}$ é o conjunto de itens e $A = \{a_1, \ldots, a_c\}$ é o conjunto de avaliações dos itens pelo usuário;
- $G_r = \{g_{r,1}, \ldots, g_{r,nc}\}$ − Conjunto de representantes dos grupos;
- p − índice do usuário ativo;
- q − índice do item para predição.

Resultado:

- $avaliacao_{est}$ − avaliação estimada

$G_{similar} \leftarrow acharGrupoMaisSimilar(p, U, G_r)$;
$avaliacao_{est} \leftarrow G_{similar}(q)$;

4.2 Resultados

A seguir apresentamos os resultados dos experimento que fizemos para avaliar o impacto potencial do algoritmo de propagação por afinidades dentro de um sistema de recomendação. Nosso primeiro experimento teve o objetivo de comparar quantitativamente os resultados do algoritmo de propagação por afinidades com o algoritmo K-means observando-se apenas o número de grupos gerados em cada um das coleções utilizadas nos experimentos. O estudo do número de grupos gerados pelo algoritmo de propagação de afinidades é importante porque o mesmo, ao contrário do K-means, não exige que o número de grupos a serem gerados seja passado como parâmetro. Tal propriedade é ao mesmo tempo uma vantagem, no sentido de que a determinação do número de grupos a serem gerados muitas vezes é deixada em segundo plano nos trabalhos que adotam recomendação baseada em agrupamento, mas por outro lado pode representar uma desvantagem, dado que o número de grupos gerados pode afetar não só a qualidade dos resultados como também o desempenho final do algoritmo de recomendação. Como parâmetro do algoritmo de propagação de afinidades é passado o fator de amortecimento, que visa apresentar as mudanças que podemos ajustar para que o algoritmo possa encontrar a convergência mais rapidamente, também podendo influenciar na quantidade de grupos gerados.

As tabelas 4.2 e 4.3 apresentam os números de grupos obtidos com o algoritmo de propagação de afinidades quando agrupando a matriz de usuário por usuário (Tabela 4.2) e item a item (Tabela 4.3). Os números de grupos gerados pelo algoritmo em ambos os casos são compatíveis com os números de grupos tipicamente adotados e avaliados como bons em trabalhos encontrados na literatura para todas as coleções experimentadas. Esse resultado inicial é importante porque mostra que o uso da propagação por afinidade é capaz de selecionar grupos em quantidades razoáveis sem a necessidade de que usuários tenham que intervir manualmente para conduzir o processo, o que na prática ocorre com o algoritmo K-means. Essa vantagem já era naturalmente esperada, mas apresentamos os experimentos por questão de completude. Finalmente, ao analizar os grupos formados, percebe-se que há uma relação de coesão razoável nos grupos formados, o que é um bom fator para indicar que os mesmos tem qualidade.

Tabela 4.2: Número de grupos gerados pelo agrupamento de coleções por Propagação de Afinidades baseado em usuário ao utilizar as diversas funções de similaridade estudadas

		Fator de Amortecimento				
		0.5	0.6	0.7	0.8	0.9
Coleção	Função de Similaridade	Número de Grupos (k)				
Comoda	Euclidiana	44	44	44	45	45
	Cosseno	44	44	44	45	45
	Correlação	25	25	25	25	26
Ml1m	Euclidiana	1306	1306	1306	1306	1306
	Cosseno	293	294	295	296	296
	Correlação	315	314	314	314	314
Ml2k	Euclidiana	450	450	450	450	450
	Cosseno	199	200	200	201	201
	Correlação	160	162	160	162	162
Ml100k	Euclidiana	210	210	210	210	210
	Cosseno	52	52	52	52	51
	Correlação	60	60	59	60	60

O indicativo de que o algoritmo de propagação por afinidades produz grupos com qualidade é importante, mas o principal objetivo da aplicação do algoritmo aqui é verificar sua viabilidade como algoritmo de agrupamento de base para a geração de recomendações. Para tanto, realizamos experimentos para avaliar as recomendações

Tabela 4.3: Número de grupos obtidos com o agrupamento de coleções por Propagação de Afinidades baseado em item ao utilizar as diversas funções de similaridade estudadas

		Fator de Amortecimento				
		0.5	0.6	0.7	0.8	0.9
Coleção	Função de Similaridade	Número de Grupos (k)				
Comoda	Euclidiana	642	472	268	269	269
	Cosseno	339	306	262	159	159
	Correlação	150	160	160	83	86
Ml1m	Euclidiana	977	965	965	964	965
	Cosseno	331	326	326	326	326
	Correlação	345	344	344	346	346
Ml100k	Euclidiana	512	489	463	462	462
	Cosseno	158	159	157	158	158
	Correlação	161	162	161	160	161

geradas pelos dois algoritmos de recomendação estudados ao usar várias funções de similaridade entre objetos e ao usar o algoritmo K-means e o algoritmo de propagação de afinidades. As tabelas 4.4 e 4.5 apresentam os resultados obtidos com os algoritmos Propagação de Afinidades e K-means quando avaliados em função da métrica RMSE e MAE tanto para a recomendação baseada em agrupamento de itens quanto para a baseada em agrupamento de usuários nas diversas coleções experimentadas. Os valores sem resultado identificados como "-" significam que os resultados não foram obtidos em tempo hábil, levando muito tempo (mais de duas semanas) para sua conclusão.

Com relação aos resultado, a primeira observação importante é que os resultados obtidos com o algoritmo de Propagação de Afinidades são competitivos quando comparados aos obtidos com o K-means, sendo superiores em alguns casos e inferiores em outros. Tal resultado se mostra importante, dado que o K-means é praticamente um padrão adotado em métodos de recomendação baseados em agrupamento. Tal resultado é um indicador forte de que novas linhas de pesquisa visando explorar o uso de algoritmos de Propagação de Afinidades podem resultar em avanços na área de recomendação baseada em agrupamento. Quando considerando os resultados com agrupamento baseado em usuários, o algoritmo de propagação de afinidades foi inferior apenas na base de dados Comoda e obteve resultados superiores nos demais casos. Considerando-se o

agrupamento baseado em itens, o algoritmo de Propagação por Afinidades foi superior em todos os casos.

Tabela 4.4: Resultados utilizando o método de agrupamento Propagação de Afinidades - Técnica de Geração de Representantes de Grupos baseada na Frequência das Avaliações

			Fator de Amortecimento									
			0.5		0.6		0.7		0.8		0.9	
		Sim.	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Comoda	Usuário	Euclidiana	1,43	0,52	1,42	0,51	1,42	0,51	1,45	0,53	1,48	0,55
		Cosseno	1,43	0,53	1,43	0,52	1,44	0,52	1,42	0,51	1,42	0,51
		Correlação	1,35	0,46	1,37	0,47	1,39	0,48	1,43	0,50	1,37	0,47
	Item	Euclidiana	1,02	0,31	0,95	0,27	0,94	0,26	0,98	0,27	1,01	0,29
		Cosseno	1,50	0,79	1,44	0,77	1,47	0,82	1,65	0,98	1,65	0,98
		Correlação	1,58	0,91	1,53	0,87	1,53	0,87	1,63	0,97	1,59	0,94
Ml1m	Usuário	Euclidiana	1,11	0,45	1,12	0,45	1,11	0,45	1,11	0,45	1,11	0,45
		Cosseno	1,72	1,04	1,71	1,04	1,71	1,04	1,71	1,04	1,71	1,04
		Correlação	1,79	1,07	1,79	1,07	1,79	1,07	1,80	1,07	1,80	1,07
	Item	Euclidiana	0,88	0,29	0,88	0,29	0,88	0,29	0,87	0,29	0,87	0,29
		Cosseno	1,70	0,97	1,70	0,97	1,70	0,97	1,70	0,97	1,70	0,97
		Correlação	1,77	0,98	1,76	0,98	1,76	0,97	1,77	0,98	1,77	0,98
Ml2k	Usuário	Euclidiana	1,00	0,40	1,00	0,40	1,00	0,41	1,00	0,41	1,00	0,40
		Cosseno	1,57	0,91	1,57	0,91	1,56	0,91	1,57	0,91	1,57	0,91
		Correlação	1,61	0,93	1,61	0,93	1,61	0,93	1,60	0,92	1,60	0,92
Ml100k	Usuário	Euclidiana	1,34	0,59	1,33	0,59	1,33	0,59	1,33	0,60	1,33	0,60
		Cosseno	1,80	1,12	1,80	1,12	1,80	1,12	1,80	1,12	1,80	1,12
		Correlação	1,89	1,16	1,88	1,15	1,87	1,14	1,88	1,14	1,88	1,15
	Item	Euclidiana	0,93	0,29	0,93	0,30	0,92	0,29	0,93	0,30	0,94	0,30
		Cosseno	1,66	0,96	1,66	0,96	1,65	0,96	1,65	0,95	1,65	0,96
		Correlação	1,78	1,00	1,78	1,00	1,78	1,00	1,78	1,00	1,78	1,00

Tabela 4.5: Resultados utilizando o método de agrupamento K-means- Técnica de Geração de Representantes de Grupos baseada na Frequência das Avaliações

			Número de Grupos									
			2		5		10		20		50	
		Sim.	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Comoda	Usuário	Euclidiana	1,97	1,06	1,82	0,88	1,89	0,94	1,70	0,75	0,91	0,21
		Cosseno	1,96	1,06	1,96	1,01	1,98	1,02	1,63	0,69	0,90	0,21
		Correlação	1,96	1,05	1,90	0,97	1,72	0,77	1,68	0,73	0,95	0,23
	Item	Euclidiana	1,42	0,93	1,47	0,94	1,50	0,94	1,53	0,94	1,61	0,96
		Cosseno	1,41	0,91	1,52	0,97	1,38	0,87	1,69	1,04	1,57	0,94
		Correlação	1,39	0,90	1,54	0,99	1,57	0,98	1,59	0,98	1,57	0,94
Ml1m	Usuário	Euclidiana	-	-	-	-	-	-	-	-	-	-
		Cosseno	-	-	-	-	-	-	-	-	-	-
		Correlação	-	-	-	-	-	-	-	-	-	-
	Item	Euclidiana	1,29	0,87	1,32	0,85	1,39	0,89	1,42	0,91	1,47	0,93
		Cosseno	1,31	0,87	1,32	0,86	1,37	0,87	1,44	0,89	1,59	0,95
		Correlação	1,29	0,86	1,37	0,88	1,42	0,90	1,54	0,95	1,65	0,97
Ml2k	Usuário	Euclidiana	1,08	0,70	1,16	0,73	1,27	0,78	1,42	0,85	1,55	0,91
		Cosseno	1,11	0,71	1,20	0,75	1,31	0,80	1,40	0,83	1,58	0,92
		Correlação	1,14	0,73	1,22	0,76	1,33	0,80	1,43	0,85	1,62	0,94
Ml100k	Usuário	Euclidiana	1,38	0,92	1,51	0,99	1,59	1,03	1,70	1,08	1,84	1,15
		Cosseno	1,35	0,91	1,44	0,94	1,57	1,00	1,71	1,08	1,85	0,14
		Correlação	1,36	0,91	1,47	0,96	1,60	1,03	1,74	1,10	1,87	1,15
	Item	Euclidiana	1,31	0,88	1,31	0,86	1,35	0,87	1,45	0,91	1,45	0,91
		Cosseno	1,37	0,90	1,39	0,89	1,41	0,89	4,07	3,93	4,18	4,04
		Correlação	1,38	0,91	1,39	0,91	1,45	0,92	1,51	0,94	1,66	0,99

As tabelas 4.6 e 4.7 apresentam os resultados obtidos quando aplicamos os dois algoritmos de agrupamento combinados à técnica de recomendação baseada na média das notas mais altas e mais baixas. Nessa abordagem de recomendação o uso dos dois algoritmos mostrou-se praticamente indiferente, com os melhores resultados sendo pra-

ticamente os mesmos tanto em RMSE quanto em MAE em tanto para a recomendação com agrupamento baseado em itens quanto para a recomendação com agrupamento baseado em usuários. Um ponto negativo no caso do algoritmo de Propagação por Afinidades foi o fato de seus melhores resultados terem sido obtidos com a função de similaridade Euclidiana, a qual tem custo computacional ligeiramente maior que as demais.

Tabela 4.6: Resultados utilizando o método de agrupamento Propagação de Afinidades - Técnica de Geração de Representantes de Grupos baseada nas Médias das Maiores e Menores Avaliações

			Fator de Amortecimento									
			0.5		0.6		0.7		0.8		0.9	
		Sim.	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Comoda	Usuário	Euclidiana	0,34	0,13	0,33	0,12	0,32	0,12	0,34	0,13	0,34	0,12
		Cosseno	0,32	0,12	0,32	0,12	0,33	0,12	0,33	0,13	0,33	0,13
		Correlação	0,29	0,10	0,29	0,10	0,29	0,10	0,29	0,10	0,28	0,09
	Item	Euclidiana	0,33	0,15	0,31	0,14	0,29	0,11	0,30	0,12	0,29	0,11
		Cosseno	0,75	0,50	0,78	0,53	0,81	0,56	0,86	0,63	0,87	0,64
		Correlação	0,84	0,60	0,84	0,60	0,84	0,60	0,89	0,65	0,89	0,65
Ml1m	Usuário	Euclidiana	0,59	0,30	0,59	0,30	0,59	0,30	0,59	0,30	0,59	0,30
		Cosseno	0,83	0,63	0,84	0,63	0,84	0,63	0,84	0,63	0,84	0,63
		Correlação	0,80	0,59	0,80	0,59	0,80	0,59	0,80	0,59	0,80	0,59
	Item	Euclidiana	0,46	0,18	0,46	0,18	0,46	0,19	0,46	0,19	0,47	0,19
		Cosseno	0,75	0,54	0,75	0,54	0,75	0,54	0,75	0,54	0,75	0,54
		Correlação	0,69	0,48	0,69	0,48	0,69	0,48	0,69	0,47	0,69	0,48
Ml2k	Usuário	Euclidiana	0,57	0,29	0,57	0,29	0,57	0,29	0,57	0,29	0,57	0,29
		Cosseno	0,77	0,57	0,77	0,57	0,77	0,57	0,77	0,57	0,77	0,57
		Correlação	0,75	0,55	0,76	0,55	0,75	0,55	0,76	0,55	0,76	0,55
Ml100k	Usuário	Euclidiana	0,62	0,33	0,62	0,32	0,61	0,32	0,61	0,33	0,61	0,33
		Cosseno	0,86	0,65	0,87	0,66	0,87	0,66	0,86	0,65	0,86	0,65
		Correlação	0,81	0,60	0,81	0,60	0,81	0,60	0,81	0,60	0,81	0,60
	Item	Euclidiana	0,42	0,16	0,42	0,16	0,42	0,16	0,42	0,16	0,42	0,16
		Cosseno	0,78	0,57	0,79	0,57	0,79	0,57	0,78	0,57	0,79	0,57
		Correlação	0,71	0,49	0,71	0,49	0,71	0,50	0,71	0,50	0,71	0,50

Tabela 4.7: Resultados utilizando o método de agrupamento K-means - Técnica de Geração de Representantes de Grupos baseada na Médias das Maiores e Menores Avaliações

			Número de Grupos									
			2		5		10		20		50	
		Sim.	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Comoda	Usuário	Euclidiana	0,62	0,38	0,52	0,28	0,51	0,27	0,46	0,22	0,20	0,47
		Cosseno	0,62	0,38	0,56	0,32	0,53	0,29	0,40	0,18	0,24	0,05
		Correlação	0,61	0,37	0,56	0,32	0,45	0,22	0,45	0,21	0,22	0,05
	Item	Euclidiana	0,99	0,78	0,98	0,76	0,96	0,73	0,93	0,70	0,89	0,67
		Cosseno	1,00	0,79	0,99	0,77	0,97	0,74	0,93	0,70	3,90	0,67
		Correlação	1,00	0,79	0,99	0,77	0,96	0,74	0,93	0,70	0,89	0,67
Ml1m	Usuário	Euclidiana	-	-	-	-	-	-	-	-	-	-
		Cosseno	-	-	-	-	-	-	-	-	-	-
		Correlação	-	-	-	-	-	-	-	-	-	-
	Item	Euclidiana	1,32	1,01	1,34	0,97	1,33	0,94	1,32	0,94	1,34	0,97
		Cosseno	1,30	0,99	1,25	0,95	1,23	0,91	1,23	0,89	1,25	0,87
		Correlação	1,26	0,94	1,27	0,95	1,26	0,94	1,27	0,91	1,28	0,88
Ml2k	Usuário	Euclidiana	1,12	0,84	1,14	0,84	1,15	0,84	1,18	0,84	1,22	0,83
		Cosseno	1,12	0,84	1,13	0,84	1,15	0,83	1,17	0,83	1,21	0,82
		Correlação	1,13	0,85	1,14	0,84	1,16	0,84	1,18	0,83	1,23	0,83
Ml100k	Usuário	Euclidiana	1,00	0,79	0,97	0,78	0,95	0,75	0,92	0,72	0,87	0,66
		Cosseno	1,00	0,79	0,97	0,77	0,94	0,74	0,90	0,70	0,84	0,63
		Correlação	1,00	0,79	0,97	0,77	0,94	0,74	0,90	0,70	0,83	0,62
	Item	Euclidiana	1,02	0,81	0,99	0,78	0,96	0,76	0,94	0,73	0,93	0,72
		Cosseno	1,00	0,80	0,95	0,75	0,93	0,73	3,98	3,82	4,15	4,02
		Correlação	1,02	0,81	0,99	0,78	0,94	0,74	0,91	0,71	3,69	3,50

As tabelas 4.8 e 4.9 apresentam os resultados obtidos quando aplicamos a abordagem baseada na média simples das notas dadas pelos usuários nos grupos representantes. Essa terceira abordagem mostrou-se bastante superior às demais em nossos experimentos, além de perceber o potencial para a aplicação do algoritmo de propagação de afinidades na tarefa de recomendação. Novamente, os resultados obtidos para a coleção Comoda com agrupamento por usuários foram inferiores aos obtidos pelo K-means se considerarmos o RMSE. Já ao considerarmos a métrica MAE, o algoritmo de propagação por afinidades foi superior até mesmo nesse caso. Nos demais casos, o algoritmo de propagação por afinidades foi muito superior ao K-means em todas as coleções e com todos os tamanhos de grupos experimentados.

Em todos os resultados percebe-se um problema prático na aplicação do algoritmo K-means, seus resultados variam bastante em função do número de grupos adotados, um parâmetro que no caso do K-means precisa ser definido a priori. Essa grande variação nos resultados mostra que uma escolha ruim no número de agrupamentos pode levar a um desempenho bastante ruim no processo de recomendação. O algoritmo de propagação de Afinidades por sua vez apresentam a vantagem de escolher automaticamente o número de grupos a serem criados. Além disso, o algoritmo mostrou pouca variação na qualidade dos grupos gerados em função do Fator de Amortecimento escolhido. Os resultados portanto mostram uma maior robustez do algoritmo quando comparado ao K-means. Outra observação importante é que o K-means só obteve resultados superiores quando aplicado à coleção Comoda e mesmo assim apenas no agrupamento baseado em usuários. O que percebemos nesse caso é que a coleção é muito pequena e esparsa, principalmente quando se tenta fazer agrupamento por usuários. Nessa situação o algoritmo de propagação por afinidades não obteve resultados superiores.

Após a apresentação dos dados com os diversos parâmetros experimentados em cada algoritmo, é interessante compararmos os melhores resultados em termos de RMSE e MAE para os métodos em todo o experimento realizado. A tabela 4.10 apresenta um resumo dos experimentos discutidos no capítulo onde são mostrados apenas os

Tabela 4.8: Resultados utilizando o métodos de agrupamento K-means - Técnica de Geração de Representantes de Grupos baseada na Média das Avaliações

			Número de Grupos									
			2		5		10		20		50	
		Sim.	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Comoda	Usuário	Euclidiana	0,62	0,38	0,52	0,28	0,51	0,27	0,46	0,22	0,20	0,47
		Cosseno	0,62	0,38	0,56	0,32	0,53	0,29	0,40	0,18	0,24	0,05
		Correlação	0,61	0,37	0,56	0,32	0,45	0,22	0,45	0,21	0,22	0,05
	Item	Euclidiana	0,99	0,78	0,98	0,76	0,96	0,73	0,93	0,70	0,89	0,67
		Cosseno	1,00	0,79	0,99	0,77	0,97	0,74	0,93	0,70	3,90	0,67
		Correlação	1,00	0,79	0,99	0,77	0,96	0,74	0,93	0,70	0,89	0,67
Ml1m	Usuário	Euclidiana	-	-	-	-	-	-	-	-	-	-
		Cosseno	-	-	-	-	-	-	-	-	-	-
		Correlação	-	-	-	-	-	-	-	-	-	-
	Item	Euclidiana	1,02	0,99	0,99	0,79	0,97	0,76	0,91	0,72	0,87	0,65
		Cosseno	1,01	0,81	0,96	0,76	0,93	0,73	0,89	0,69	0,83	0,64
		Correlação	0,99	0,79	0,97	0,77	0,94	0,74	0,91	0,71	0,85	0,64
Ml2k	Usuário	Euclidiana	0,88	0,68	0,87	0,67	0,86	0,66	0,84	0,64	0,80	0,60
		Cosseno	0,88	0,68	0,87	0,67	0,85	0,65	0,82	0,63	0,77	0,58
		Correlação	0,88	0,68	0,87	0,67	0,84	0,65	0,82	0,62	0,77	0,57
Ml100k	Usuário	Euclidiana	1,00	0,79	0,97	0,78	0,95	0,75	0,92	0,72	0,87	0,66
		Cosseno	1,00	0,79	0,97	0,77	0,94	0,74	0,90	0,70	0,84	0,63
		Correlação	1,00	0,79	0,97	0,77	0,94	0,74	0,90	0,70	0,83	0,62
	Item	Euclidiana	1,02	0,81	0,99	0,78	0,96	0,76	0,94	0,73	0,93	0,72
		Cosseno	1,00	0,80	0,95	0,75	0,93	0,73	3,98	3,82	4,15	4,02
		Correlação	1,02	0,81	0,99	0,78	0,94	0,74	0,91	0,71	3,69	3,50

Tabela 4.9: Resultados utilizando o método de agrupamento Propagação de Afinidades - Técnica de Geração de Representantes de Grupos baseada na Média das Avaliações

			Fator de Amortecimento									
			0.5		0.6		0.7		0.8		0.9	
		Sim.	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Comoda	Usuário	Euclidiana	0,34	0,13	0,33	0,12	0,32	0,12	0,34	0,13	0,34	0,12
		Cosseno	0,32	0,12	0,32	0,12	0,33	0,12	0,33	0,13	0,33	0,13
		Correlação	0,29	0,10	0,29	0,10	0,29	0,10	0,29	0,10	0,28	0,09
	Item	Euclidiana	0,33	0,15	0,31	0,14	0,29	0,11	0,30	0,12	0,29	0,11
		Cosseno	0,75	0,50	0,78	0,53	0,81	0,56	0,86	0,63	0,87	0,64
		Correlação	0,84	0,60	0,84	0,60	0,84	0,60	0,89	0,65	0,89	0,65
Ml1m	Usuário	Euclidiana	0,59	0,30	0,59	0,30	0,59	0,30	0,59	0,30	0,59	0,30
		Cosseno	0,83	0,63	0,84	0,63	0,84	0,63	0,84	0,63	0,84	0,63
		Correlação	0,80	0,59	0,80	0,59	0,80	0,59	0,80	0,59	0,80	0,59
	Item	Euclidiana	0,46	0,18	0,46	0,18	0,46	0,19	0,46	0,19	0,47	0,19
		Cosseno	0,75	0,54	0,75	0,54	0,75	0,54	0,75	0,54	0,75	0,54
		Correlação	0,69	0,48	0,69	0,48	0,69	0,48	0,69	0,47	0,69	0,48
Ml2k	Usuário	Euclidiana	0,57	0,29	0,57	0,29	0,57	0,29	0,57	0,29	0,57	0,29
		Cosseno	0,77	0,57	0,77	0,57	0,77	0,57	0,77	0,57	0,77	0,57
		Correlação	0,75	0,55	0,76	0,55	0,75	0,55	0,76	0,55	0,76	0,55
Ml100k	Usuário	Euclidiana	0,62	0,33	0,62	0,32	0,61	0,32	0,61	0,33	0,61	0,33
		Cosseno	0,86	0,65	0,87	0,66	0,87	0,66	0,86	0,65	0,86	0,65
		Correlação	0,81	0,60	0,81	0,60	0,81	0,60	0,81	0,60	0,81	0,60
	Item	Euclidiana	0,42	0,16	0,42	0,16	0,42	0,16	0,42	0,16	0,42	0,16
		Cosseno	0,78	0,57	0,79	0,57	0,79	0,57	0,78	0,57	0,79	0,57
		Correlação	0,71	0,49	0,71	0,49	0,71	0,50	0,71	0,50	0,71	0,50

melhores resultados obtidos para o K-means e para o algoritmo de Propagação por Afinidades. Pode-se ver que quando levamos em conta os melhores parâmetros para cada método, o algoritmo K-means obteve resultados superiores ao de Propagação de Afinidades apenas na coleção Comoda. Nos demais casos, o algoritmo de Propagação de Afinidades mostrou-se uma excelente alternativa para ser usada como parte de um processo de recomendação baseada em agrupamentos.

Tabela 4.10: Resumo dos Melhores Resultados Apresentados nos Experimentos

	Propagação de Af.		K-means	
	RMSE	MAE	RMSE	MAE
Comoda	0,28	0,09	0,20	0,05
Ml1m	0,46	0,18	0,83	0,64
Ml2k	0,57	0,29	0,82	0,62
Ml100k	0,42	0,16	0,83	0,62

Capítulo 5

Conclusão

Este trabalho apresentou um sistema de recomendação com abordagem baseada em agrupamentos usando o método de agrupamento Propagação de Afinidades(PA). Diferentemente da maioria dos métodos convencionais de agrupamento, PA considera inicialmente todos os pontos de amostra como possíveis exemplares, realizando troca de mensagens determinísticas até emergir gradualmente o conjunto de exemplares. PA ganhou grande popularidade na aplicação em áreas da bioinformática, apresentando bons resultados para problemas de agrupamentos de sequências de DNA, de faces (imagem), de coleções de filmes, e na sumarização de textos. O método foi implementado neste trabalho com a expectativa de seguir com bons resultados na área de sistemas de recomendação utilizando a abordagem baseada em agrupamentos.

Na abordagem utilizando agrupamento, o método de particionamento PA foi usado na fase *offline* para identificar usuários/itens similares. Os objetos mais similares foram organizados em grupos e representados por um vetor de valores representantes. Os representantes dos agrupamentos foram calculados baseados nas frequências de notas/avaliações pertencentes aos usuários daquele grupo.

Na fase *online* foi necessário criar uma lista de recomendações para o usuário ativo, que ao invés de realizar comparações com todos os dados arquivados, realizou apenas comparações com o representante do grupo ao qual faz parte, diminuindo consideravelmente a quantidade de cálculos de distância e operações de comparação nessa fase.

A abordagem apresentada neste documento pode ser caracterizada por baixa pre-

cisão se comparada com as melhores abordagens na literatura, entretanto, com o tempo de resposta bom e escalável, permite ser utilizada em aplicações de tempo real.

Em termos de comparação de resultados, a função de similaridade que apresentou melhores resultados usando PA foi a euclidiana, na recomendação baseada em agrupamento de itens. Chegou a diminuir em torno da metade de RMSE na coleção *Ml100k* em seu melhor resultado se comparado com o melhor resultado apresentado utilizando o *k-means*.

Os experimentos realizados neste documento mostraram que o sistema de recomendação baseado na técnica de agrupamento PA conseguiu melhores resultados em vários cenários comparados com a técnica de agrupamento K-means. Um fator importante para o melhor desempenho do PA foi a coesão dos itens aglomerados, demonstrando qualidade na tarefa de agrupamento, outro fator que deve ser destacado é a estabilidade do algoritmo, por não necessitar de definição de quantidades de grupo *a priori*, como ocorre com K-means.

Para os resultados na recomendação baseada em agrupamento de usuário, a hipótese de que os resultados do algoritmo PA não tenham conseguido melhorar os resultados apresentados utilizando o *k-means* é baseada no fato do PA gerar grupos automaticamente, levando em consideração que a técnica de representantes de grupos foi criada usando *k-means* como referência, portanto, o PA pode não tirar proveito dessas características, já que não define a quantidade de grupos *a priori*, mas que por outro lado cria grupos comprovadamente de qualidade.

Neste trabalho foi apresentada uma investigação entre várias que ainda podem ser feitas no campo de pesquisa utilizando métodos de agrupamento. Existe espaço para o desenvolvimento de abordagens ou criação técnicas para melhorar resultados utilizando PA, ou ainda, implementar outros métodos de agrupamento como DBSCAN, EM, VSH(*Vertex substitution heuristic*) e investigar seus resultados de forma semelhante ao realizado com PA.

Referências Bibliográficas

Amatriain, X. (2013). Big & personal: Data and models behind netflix recommendations. In *Proceedings of the 2Nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, BigMine '13, pages 1–6, New York, NY, USA. ACM.

Bobadilla, J., Ortega, F., Hernando, A., and GutiéRrez, A. (2013). Recommender systems survey. *Know.-Based Syst.*, 46:109–132.

Bodenhofer, U., Kothmeier, A., and Hochreiter, S. (2011). Apcluster: an r package for affinity propagation clustering. *Bioinformatics*, 27(17):2463–2464.

Dueck, D. and Frey, B. J. (2007). Non-metric affinity propagation for unsupervised image categorization. In *Proceedings of the IEEE International Conference on Computer Vision*.

Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315(5814):972–976.

Frey, B. J. and Dueck, D. (2008). Response to comment on "clustering by passing messages between data points". *Science*, 319(5864):726–726.

Harper, F. M. and Konstan, J. A. (2015). The movielens datasets: History and context. In *ACM Transactions on Interactive Intelligent Systems (TiiS)*. ACM.

Kuzelewska, U. (2014). Clustering algorithms in hybrid recommender system on movielens data. *Studies in Logic, Grammar and Rhetoric*, 37(1):125–139.

Lacerda, A. and Ziviani, N. (2013). Building user profiles to improve user experience in recommender systems. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 759–764, New York, NY, USA. ACM.

Li, Q. and Kim, B. M. (2003). Clustering approach for hybrid recommender system. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 33–38.

Liao, C.-L. and Lee, S.-J. (2016). A clustering based approach to improving the efficiency of collaborative filtering recommendation. *Electron. Commer. Rec. Appl.*, 18(C):1–9.

Odic, A., Tkalcic, M., Tasic, J. F., and Kosir, A. (2013). Predicting and detecting the relevant contextual information in a movie-recommender system. *Interacting with Computers*, 25(1):74–90.

Research, G. (2017). Grouplens research. https://www.grouplens.org.

Ribeiro, M. T., Ziviani, N., Moura, E. S. D., Hata, I., Lacerda, A., and Veloso, A. (2014). Multiobjective pareto-efficient approaches for recommender systems. *ACM Trans. Intell. Syst. Technol.*, 5(4):53:1–53:20.

Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. (2010). *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition.

Rongfei, J., Maozhong, J., and Chao, L. (2010). A new clustering method for collaborative filtering. In *2010 International Conference on Networking and Information Technology*, pages 488–492.

Shi, Y., Larson, M., and Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4:2–4:2.

Tacchini, E. and Damiani, E. (2011). What is a "musical world"? an affinity propagation approach. In *Proceedings of the 1st International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies*, MIRUM '11, pages 57–62, New York, NY, USA. ACM.

Ungar, L. and P. Foster, D. (2000). Clustering methods for collaborative filtering.

Wei, S., Ye, N., Zhang, S., Huang, X., and Zhu, J. (2012). Collaborative filtering recommendation algorithm based on item clustering and global similarity. In *2012 Fifth International Conference on Business Intelligence and Financial Engineering*, pages 69–72.