CEFET/RJ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO

UM ESTUDO SOBRE A INTEGRAÇÃO DE ALGORITMOS DE FILTRAGEM COLABORATIVA E DETECÇÃO DE COMUNIDADES PARA MELHORIA DO DESEMPENHO DAS RECOMENDAÇÕES

Ryan Dutra de Abreu

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ, como parte dos requisitos necessários à obtenção do grau de Mestre em Ciência da Computação.

Orientadores:
Laura Silva de Assis
Douglas de Oliveira Cardoso

Rio de Janeiro,

9 de Janeiro de 2024

Um Estudo sobre a Integração de Algoritmos de Filtragem Colaborativa e Detecção de Comunidades para Melhoria do Desempenho das Recomendações

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação, Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ.

Ryan Dutra de Abreu

Aprovada por:

Prof. Laura Silva de Assis, D.Sc. (orientadora)

Prof. Douglas de Oliveira Cardoso, D.Sc. (coorientador)

Prof. Diego Nunes Brandão, D.Sc.

Prof. João Nuno Vinagre Marques da Silva, D.Sc.

Rio de Janeiro,

9 de Janeiro de 2024

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

A162 Abreu, Ryan Dutra de
Um estudo sobre a integração de algoritmos de filtragem colaborativa e detecção de comunidades para melhoria do desempenho das recomendações / Ryan Dutra de Abreu. — 2023.
60f. : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2023.
Bibliografia : f. 49-60
Orientadora: Laura Silva de Assis
Coorientador: Douglas de Oliveira Cardoso

1. Aprendizado de máquina. 2. Filtragem Colaborativa. 3. Redes Sociais. 4. Algoritmos. I. Assis, Laura Silva de (Orient.). II. Cardoso, Douglas de Oliveira. (Coorient.). III. Título.

CDD 006.31

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

RESUMO

Um Estudo sobre a Integração de Algoritmos de Filtragem Colaborativa e Detecção de Comunidades para Melhoria do Desempenho das Recomendações

Ryan Dutra de Abreu

Orientadores:
Laura Silva de Assis
Douglas de Oliveira Cardoso

Projeto de Pesquisa submetido ao Programa de Pós-graduação em Ciência da Computação do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ como parte dos requisitos necessários à obtenção do grau de mestre.

A recomendação de itens em sistemas de recomendação é uma técnica amplamente utilizada para auxiliar os usuários na descoberta de conteúdo relevante. Tradicionalmente, os algoritmos de recomendações baseados em filtragem colaborativa são os mais utilizados no campo devido ao seu bom desempenho frente às demais abordagens. Esses algoritmos buscam identificar padrões globais de similaridade entre usuários ou itens para gerar recomendações. Neste trabalho, investigamos como a combinação das tarefas de recomendação e detecção de comunidades pode levar a recomendações melhores do que aquelas obtidas sem considerar comunidades implícitas. Para isso, avaliamos experimentalmente várias combinações de métodos para detecção de comunidades e algoritmos de recomendação, os quais foram submetido à experimentos computacionais considerando tanto conjuntos de dados sintéticos quanto reais. O objetivo intrínseco deste esforço foi o de revelar padrões interessantes no comportamento dos sistemas resultantes. Os resultados obtidos mostram que a inclusão de detectores de comunidades no sistema pode melhorar significativamente tanto a eficácia quanto a eficiência dos algoritmos de recomendação em alguns cenários. Essas descobertas podem ser usadas para ajudar pesquisadores e profissionais de ciência de dados a compreender melhor os benefícios e limitações dessa metodologia. Por fim, as descobertas aqui obtidas podem ser aplicadas em menor ou maior grau em diversos domínios onde a personalização de recomendações em nível local pode ser uma abordagem eficaz para melhorar a experiência do usuário, sobretudo em situações onde exista uma grande estrutura de redes e comunidades bem definidas.

Palavras-chave:
Sistemas de Recomendação, Filtragem Colaborativa, Detecção de Comunidades, Redes Sociais.

Rio de Janeiro,

9 de Janeiro de 2024

ABSTRACT

Um Estudo sobre a Integração de Algoritmos de Filtragem Colaborativa e Detecção de Comunidades para Melhoria do Desempenho das Recomendações

Ryan Dutra de Abreu

Advisors:
Laura Silva de Assis
Douglas de Oliveira Cardoso

Abstract of dissertation submitted to Programa de Pós-graduação em Ciência da Computação - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca CEFET/RJ as partial fulfillment of the requirements for the degree of master.

Item recommendation in recommender systems is a widely used technique to assist users in discovering relevant content. Traditionally, collaborative filtering-based recommendation algorithms have been the most commonly used in the field due to their sound performance compared to other approaches. These algorithms aim to identify global patterns of similarity between users or items to generate recommendations. In this thesis, we investigate how combining the tasks of recommendation and community detection can lead to better recommendations than those obtained without considering implicit communities. To achieve this, we experimentally evaluate various combinations of community detection methods and recommendation algorithms, subjecting these different arrangements to both synthetic and real-world datasets. The intrinsic goal of this effort was to unveil interesting patterns in the behavior of the resulting systems. The obtained results show that the inclusion of community detectors in the system can significantly improve both the effectiveness and efficiency of recommendation algorithms in some scenarios. These findings can be used to assist researchers and data science professionals in better understanding the benefits and limitations of this methodology. Lastly, the discoveries obtained here can be applied to a greater or lesser extent in various domains where personalized recommendations at the local level can be an effective approach to enhance the user experience, particularly in situations where there exists a large network structure and well-defined communities.

Key-words:
Recommender System, Collaborative Filtering, Community Detection, Social Networks.

Rio de Janeiro,

9 de Janeiro de 2024

Lista de Figuras

II.1 Arquitetura típica de um sistema de recomendações moderno. 5

II.2 Exemplos de sistemas de *feedback* explícito. 6

II.3 Exemplo de um grafo não ponderado e não direcionado. 9

II.4 Matriz de adjacência do grafo ilustrado na Figura II.3. 9

II.5 Exemplos de um grafo bipartido com dois grupos de nós. 9

II.6 Exemplo de uma rede com três estruturas comunitárias. As conexões dentro de cada comunidade são densas e robustas, enquanto que os pontos de contato (arestas) intercomunitários são esparsos. 11

II.7 Visualização das etapas do algoritmo de Louvain. Adaptado de Blondel et al. [2008a] 13

IV.1 Exemplo de um grafo ponderado e bipartido e que representa as relações usuário-item . 27

IV.2 Projeção correspondente da rede ilustrada na Figura IV.1 27

IV.3 Projeção do grafo com duas comunidades identificadas. 27

V.1 Distribuição das Avaliações no Movielens-100k. 32

V.2 Distribuição das Avaliações no Movielens-1Mk 32

V.3 Fluxograma das etapas seguidas para criação dos dados sintéticos. 34

V.4 Desempenho em Root Mean Squared Error (RMSE) considerando diferentes conjuntos de dados, abordagens de detecção de comunidades e algoritmos de recomendação. 38

V.5 Diferença Percentual (DP) entre o RMSE da abordagem proposta e o da abordagem padrão. 40

V.6 Distribuição dos 25 Melhores e 25 Piores casos em termos da DP de RMSE. 41

V.7 Tempo de treinamento considerando diferentes conjuntos de dados, abordagens de detecção de comunidades e algoritmos de recomendação. 42

V.8 DP entre o tempo de treino da abordagem proposta e o da abordagem padrão. 43

V.9 Distribuição dos 25 Melhores e 25 Piores casos em termos da DP do tempo de treino. 44

V.10 Tempo de predição considerando diferentes conjuntos de dados, abordagens de detecção de comunidades e algoritmos de recomendação. 45

Lista de Tabelas

III.1 Critérios de Inclusão 16

III.2 Critérios de Exclusão 16

III.3 Lista de artigos da revisão sistemática com informações gerais resumidas. 24

V.1 Amostra de Avaliações do Movielens 100k 32

V.2 Tempo despendido no processo de detecção de comunidades. 42

Lista de Abreviações

AHC	Agglomerative Hierarchical Clustering	17, 20, 23
ARI	Adjusted Rand Index	18
DBLP	Digital Bibliographic Library Browser	16, 23
DP	Diferença Percentual	39, 40, 41, 42, 43, 44
FC	Filtragem Colaborativa	2, 3, 4, 7, 8, 16, 17, 20, 21, 22, 47
HMETIS	H-Way Multilevel Partitioning	20, 23
KMETIS	K-Way Multilevel Partitioning	20, 23
MAD	Median Absolute Deviation	42, 45
MAE	Mean Absolut Error	18, 20, 21
MINHASH	Mínima Função De Hashing	22
NDCG	Normalized Discounted Cumulative Gain	22
NMF	Non-Negative Matrix Factorization	37, 38, 39, 43, 44
PLSA	Probabilistic Latent Semantic Analysis	22, 23
RMSE	Root Mean Squared Error	18, 20, 21, 37, 38, 39, 40, 41, 45, 47
ROCK	Robust Clustering Using Links	20, 23
ROI	Return On Investments	48
SBG	Stochastic Block Model	35
SIM	Similarity-Based Model	17
SVD	Singular Value Decomposition	17, 18, 21, 23, 37, 38

Sumário

I Introdução **1**

I.1 Motivação 2

I.2 Objetivos 3

I.3 Organização dos Capítulos 3

II Fundamentação Teórica **4**

II.1 Sistemas de Recomendação 4

II.1.1 Filtragem Colaborativa Baseada em Memória 7

II.1.2 Filtragem Colaborativa Baseada em Modelo 8

II.2 Grafos e Estruturas Comunitárias 8

II.2.1 Detecção de Comunidades 10

III Trabalhos Relacionados **15**

III.1 Processo de Pesquisa 15

III.2 Revisão da Literatura 16

IV Abordagem Proposta **25**

IV.1 Projeção do Grafo Bipartido 25

IV.2 Detecção de Comunidades 26

IV.3 Instanciamento de Modelos e Treinamento 28

IV.4 Predição 29

V Avaliação Experimental **31**

V.1 Base de Dados 31

V.1.1 Bases de Dados Movielens 31

V.1.2 Bases de Dados Sintéticas 33

V.2 Protocolo Experimental 36

V.3 Resultados e Discussões 38

V.3.1 Desempenho 38

V.3.2 Custo Computacional 42

VI Considerações Finais **47**

Referências 49

Capítulo I Introdução

O grande crescimento da internet experimentado nos últimos trinta anos mudou de maneira permanente a forma como a sociedade realiza negócios e se comunica [Castells, 2004]. Se no início da década de 1990 a participação da população com acesso à rede global de computadores não chegava a 1% – mesmo para os países desenvolvidos do bloco comum europeu ou da América anglo-saxônica – em 2020 nove em cada dez estadunidenses (91%) já estavam *on-line*, e no mesmo período boa parte das nações alcançou níveis semelhantemente elevados de acesso. No Brasil, 81% das pessoas já utilizavam a internet; na França 85%; na Malásia 89%; na Dinamarca e Coreia do Sul 96%; na Noruega 97%; e na Islândia 99% [Roser et al., 2015]. Como resultado desse processo, a sociedade passou a adotar e integrar amplamente as novas tecnologias de informação e comunicação no seu cotidiano, dando origem às novas estruturas de relacionamento, hábitos, padrões de consumo e aos amplos e modernos mercados digitais [Paul and Aithal, 2018], onde uma infinidade de produtos e serviços pôde finalmente ser transacionada sem as antigas restrições impostas pelo mundo físico.

Entretanto, se por um lado esse processo trouxe maior disponibilidade e diversidade para o consumidor, ele também ocasionou superexposição, visualização excessiva de conteúdo e inserção abundante de dados [Gross, 1964; Speier et al., 1999; Himma and Tavani, 2008; Dias, 2014; Roetzel, 2019]. Desse modo, acentuou-se a problemática da sobrecarga informacional, descrita como a dificuldade em encontrar informações relevantes ou de se tomar decisões de maneira mais assertiva quando se tem muitas opções à disposição [Yang et al., 2003; Bawden et al., 1999; Jacoby, 1984; Goodall, 1967].

Por conseguinte, os sistemas de recomendação surgiram como uma ferramenta promissora para gerenciamento e atenuação da infoxicação [Dias, 2014]. Ao sugerir os itens que potencialmente poderiam ser mais interessantes aos usuários, os mecanismos de recomendação simplificaram o processo de escolha, tornando-se importantes ferramentas para empresas como Amazon Inc., Netflix, Youtube, *British Broadcasting Corporation* e Spotify [Golbeck, 2006; Massa and Avesani, 2007; Zhang et al., 2019; Panteli et al., 2019; Werner, 2020]. Essas companhias passaram a fornecer aos usuários recomendações personalizadas e de qualidade para itens como notícias, artigos, livros, filmes, vídeos, faixas de músicas, dentre outros. Dessa forma, tornaram-se referências de otimização da experiência do usuário em seus respectivos mercados.

I.1 Motivação

Os modernos sistemas de recomendação são um fértil campo de estudo derivado da ciência cognitiva e da recuperação informacional [Rich, 1979; Sanderson and Croft, 2012]. De modo geral, pode-se compreendê-los como uma estrutura lógica que gera recomendações de produtos ou itens que sejam potencialmente interessantes aos usuários com base em informações coletadas, como histórico de consumo, nível de preferência, dados demográficos, e outros elementos [Bobadilla et al., 2013].

Os métodos de recomendação podem ser classificados em três categorias principais: *(i)* abordagens colaborativas, *(ii)* baseadas em conteúdo, e *(iii)* de filtragem híbrida [Adomavicius and Tuzhilin, 2005; Pazzani and Billsus, 2007; Burke, 2000; Schafer et al., 2007].

As técnicas de Filtragem Colaborativa (FC) são hoje as mais utilizadas devido à sua alta precisão e eficiência. Elas geram recomendações à cada indivíduo com base nos demais usuários presentes no conjunto de dados que compartilham de gostos semelhantes. A principal ideia por trás dessas abordagens reside na hipótese de que pessoas com padrões de escolha ou consumo semelhantes muito provavelmente terão opiniões parecidas sobre um novo item ainda não conhecido por nenhuma delas [Gorripati and Vatsavayi, 2017].

No entanto, como na prática muitos sistemas de recomendação comerciais são baseados em grandes conjuntos de dados, as matrizes de usuários e/ou itens utilizadas pelos algoritmos de FC podem ser extremamente grandes e esparsas, uma vez que, em geral, os usuários ativos costumam avaliar apenas uma pequena parte dos itens.

Este fato acaba por suscitar desafios ao desempenho das recomendações fornecidas, além de acarretar problemas secundários, como o do *cold-start*. Como os métodos de FC recomendam itens com base nas preferências anteriores dos usuários, novos usuários precisarão avaliar um número suficientemente grande de itens para permitir que o sistema capture suas preferências com precisão e, assim, forneça recomendações confiáveis. Portanto, usuários que só tenham avaliado um pequeno número de itens, os *cold-starters*, tendem a prejudicar o desemprenho geral do sistema [Guo, 2013][1].

Para contornar essas questões, algumas estratégias gerais têm sido discutidas na literatura, como a incorporação de informações adicionais para modelar as preferências dos usuários, a inclusão de novas medidas de similaridade, e, sobretudo, a adoção de abordagens que utilizem algoritmos de detecção de comunidades ou agrupamento para realizar inferências no nível dessas redes locais, ao invés de em toda a amálgama de dados [Girvan and Newman, 2002; Xie and Szymanski, 2013; Guo, 2013; Gorripati and Vatsavayi, 2017; Keerthi Gorripati and Angadi, 2018; Bourhim et al., 2018b].

[1]Embora menos comum, o termo *cold-starter* também pode aplicar-se à itens que tenham sido pouco avaliados. Esses itens também tendem a prejudicar o desempenho geral dos algoritmos de FC.

I.2 Objetivos

Quando aplicados a conjuntos de dados comerciais, os algoritmos clássicos de FC muitas vezes enfrentam problemas de esparsidade. Isso significa que têm dificuldade em lidar com situações em que a quantidade de avaliações disponíveis é limitada em relação ao tamanho total dos dados. Além disso, esses algoritmos podem encontrar dificuldade em fazer recomendações para usuários que ainda não avaliaram um número suficientemente grande de itens e/ou para sugerir itens que foram raramente avaliados (*cold-starters*).

Para mitigação desses problemas, abordagens recentes de recomendação baseadas em algoritmos de detecção de comunidades ou agrupamento têm sido desenvolvidas. No entanto, embora a integração desses detectores aos sistemas de recomendação já tenha se mostrado promissora no aumento do desempenho das recomendações realizadas, ainda existem algumas limitações a serem consideradas, além de um amplo espaço para exploração e experimentação no campo de estudo, dada a incipiência de esforços anteriores nesse sentido.

Neste trabalho são ampliadas as pesquisas existentes através de uma investigação sobre como a combinação dessas tarefas pode levar a recomendações melhores do que aquelas obtidas sem a consideração de comunidades implícitas. Para tanto, versões de algoritmos de FC com e sem detectores de comunidade são comparadas em uma variedade de contextos experimentais. Ao explorar variados arranjos e avaliar sua eficácia e eficiência usando conjuntos de dados sintéticos e reais, buscamos fornecer *insights* valiosos sobre o comportamento das predições e contribuir para o desenvolvimento de sistemas de recomendação mais eficazes e personalizados para pesquisadores e profissionais de ciência de dados que atuam no campo de estudo.

I.3 Organização dos Capítulos

O Capítulo II trata dos conhecimentos teóricos necessários para o desenvolvimento do presente trabalho. O Capítulo III, por sua vez, explora a bibliografia pretérita de trabalhos relacionados, destacando especialmente os estudos que se concentram no desenvolvimento de soluções que combinam tarefas de recomendação com detecção de comunidades ou agrupamento. O Capítulo IV delimita a metodologia utilizada para a execução dos experimentos que serão realizados no Capítulo V. Finalmente, o Capítulo VI abarca as questões inerentes à pesquisa e tarefas ainda em aberto até a defesa da dissertação.

Capítulo II Fundamentação Teórica

Neste capítulo serão exploradas as bases teóricas dos sistemas de recomendação, com enorme destaque aos baseados em FC. Em seguida, explorar-se-ão os conceitos-chave de ciência de redes importantes à esse trabalho, incluindo definições dessas estruturas e aspectos relacionados ao problema de particionamento, como modularidade e algoritmos de detecção de comunidades.

II.1 Sistemas de Recomendação

Em sua formulação mais primitiva, o problema de recomendação pode ser convertido na tarefa de se estimar a utilidade percebida para os itens que ainda não foram vistos ou adquiridos por um usuário [Adomavicius and Tuzhilin, 2005; Ricci et al., 2011]. Intuitivamente, essa estimativa costuma basear-se nas avaliações dadas por este usuário a outros itens e em algumas outras informações, como dados demográficos e comportamento recente. Assim, de posse das estimativas para os itens ainda sem classificação, pode-se, portanto, recomendar ao usuário o(s) item(ns) de maior utilidade estimada.

A Figura II.1 exemplifica como essas recomendações são produzidas em um cenário tipicamente iterativo: ao navegar e interatuar na interface disponibilizada, como em um *marketplace* digital, ou em um catálogo de filmes *on-line*, o usuário produz informações suficientes sobre si que, quando levadas em consideração por um sistema de recomendação construído sobre uma combinação de regras robustas, fornece estimativas para a utilidade dos itens ainda não vistos do catálogo, e, por conseguinte, a recomendação daqueles produtos de maior utilidade estimada.

De maneira mais formal, o problema de recomendação pode ser modelado da seguinte forma [Adomavicius and Tuzhilin, 2005]: seja U o conjunto de todos os usuários e seja I o conjunto de todos os itens que podem ser recomendados; seja também γ uma função que mede a utilidade do j-ésimo item i_j para o k-ézimo usuário u_k, i.e., $\gamma : U \times I \to \mathbb{R}_+$. Então, para cada usuário $u_k \in U$, queremos escolher o item $i'_{u_k} \in I$ que maximiza a utilidade percebida, como apresentado na Equação (II.1).

$$i'_{u_k} = \arg\max_{i_j \in I} \ \gamma(u_k, i_j) \quad \forall \ u_k \in U. \tag{II.1}$$

No entanto, o problema central dos sistemas de recomendação reside em que a utilidade γ geralmente não está definida *a priori* em todo o domínio $U \times I$, mas apenas em algum subconjunto

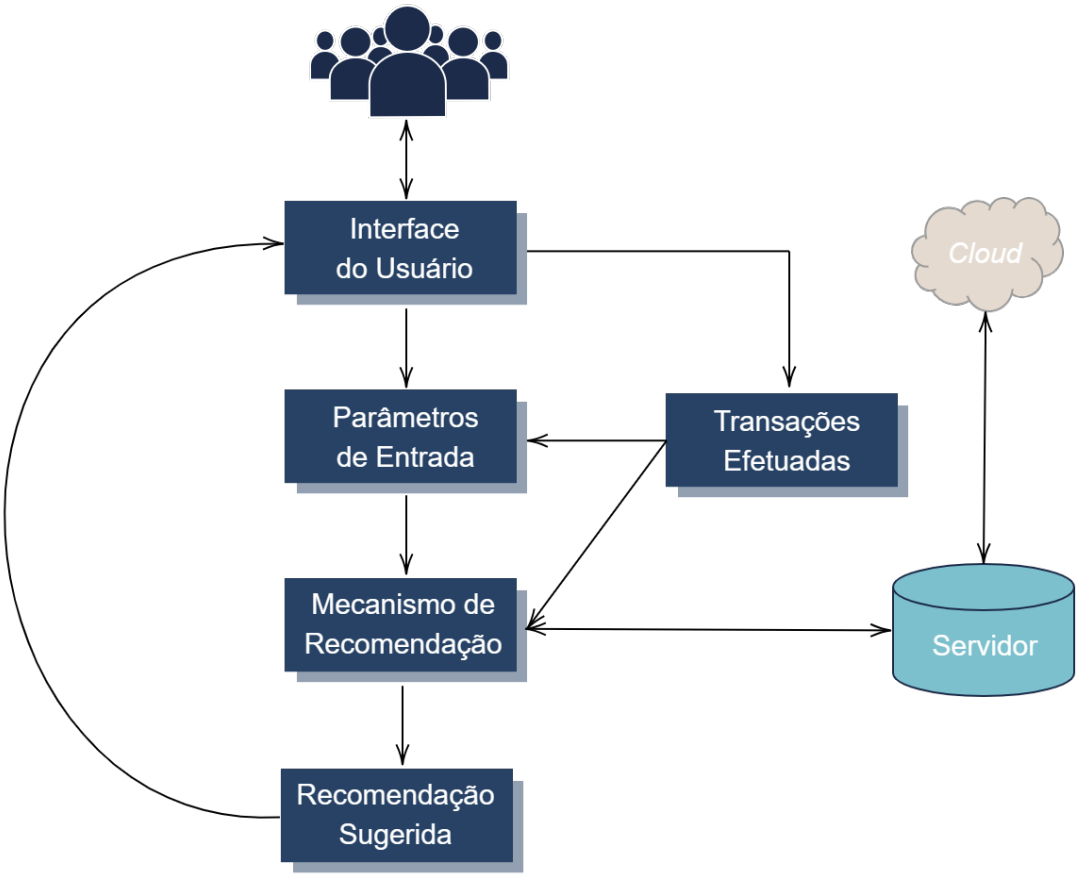


Figura II.1: Arquitetura típica de um sistema de recomendações moderno.

pequeno dele. Tal esparsidade presente na matriz (item × usuário) é um problema estrutural em sistemas de recomendação, chegando a uma proporção acima de 90% para muitos *datasets* industriais [Yildirim and Krishnamoorthy, 2008; Sharma and Gera, 2013; Mishra et al., 2017; Al-Bakri and Hashim, 2018]. Isso significa que γ precisa ser extrapolada para todo o domínio $U \times I$. De acordo com Adomavicius and Tuzhilin [2005, 2001], uma vez que essas notas desconhecidas são estimadas, as recomendações para um usuário são feitas selecionando-se os itens com estimativas mais altas entre todos as opções disponíveis.

A utilidade observada no subconjunto populado do domínio $U \times I$ que serve de base para formulação do estimador, por sua vez, é geralmente representada por uma nota que indica o quanto um usuário em particular gosta de um determinado item. Essas notas são costumeiramente fornecidas através de *feedback* explícito, como por exemplo, os realizados nas plataformas da Amazon e Netflix, como ilustrado na Figura II.2, onde as áreas em destaque ilustram o mecanismo à partir do qual o usuário pode avaliar o item em questão. No caso da Amazon, por exemplo, cada usuário é convidado a avaliar os itens com uma nota de 1 a 5, sendo 5 igual a "alta utilidade", ou "extremamente satisfeito" [Linden et al., 2003]. No caso da Netflix, por outro lado, existem apenas três opções

para indicar a utilidade percebida pelo usuário: “Amei”, “Gostei” e “Não é para mim” [Bennett and Lanning, 2007].

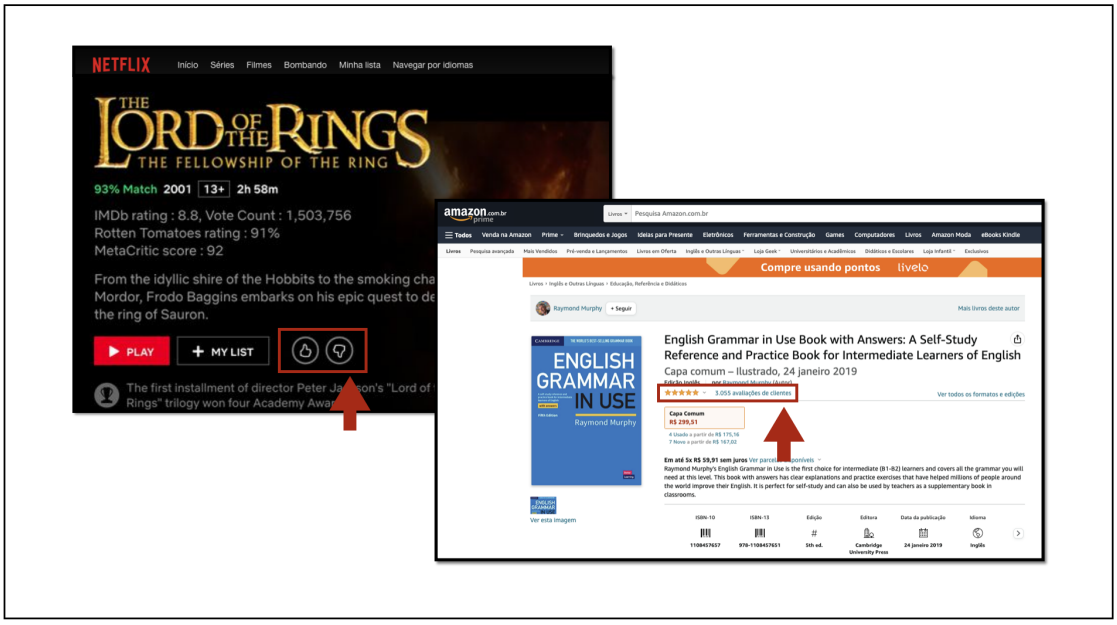


Figura II.2: Exemplos de sistemas de *feedback* explícito.

Alguns sistemas, por outro lado, se valem de métodos de determinação de notas não intrusivos, onde certas *proxies* são utilizadas para se estimar a utilidade do item, o que caracteriza um *feedback* implícito. Por exemplo, a quantidade de tempo que um usuário gasta lendo um artigo em específico pode servir como um *proxy* da utilidade que o usuário percebe nesse artigo. Outras métricas comumente utilizadas para obtenção de *feedback* implícito incluem a preparação para uso futuro, como a inclusão de filmes, músicas e artigos em alguma lista própria para posterior consumo; e o número de interações realizadas com o item, como frequência de consumo, cliques, buscas, adição ao “carrinho”, e número de vezes em que o usuário iniciou o processo de *check-out* daquele item [Konstan et al., 1997; Oard et al., 1998; Schein et al., 2002; Middleton et al., 2004; Anyosa et al., 2018; Wang et al., 2018; Jannach et al., 2018]. No entanto, esses *feedbacks* implícitos costumam ser imprecisos e não podem substituir totalmente as classificações explícitas fornecidas pelo usuário sem alguma perda de qualidade [Adomavicius and Tuzhilin, 2005].

A despeito do tipo de feedback empregado pelo sistema, as abordagens utilizadas para estimar as utilidades a partir de um subconjunto em $U \times I$ são basicamente três [Balabanović and Shoham, 1997; Burke, 2002; Adomavicius and Tuzhilin, 2005; Lu et al., 2015; Chen et al., 2015; Yang et al., 2016; He et al., 2016]:

1. **Recomendações baseadas em conteúdo:** o usuário recebe recomendações de itens semelhantes àqueles que ele preferiu no passado. A ideia por trás desse método reside na hipótese de que as preferências dos consumidores apresentam pouca volatilidade, não se modificando

abruptamente ao longo do tempo. Para tanto, um amplo conhecimento de domínio sobre os usuários e os itens é utilizado para adequação do sistema.

2. **Recomendações baseadas em FC:** o usuário recebe recomendações com base em suas próprias preferências passadas, assim como nas de muitos outros usuários (colaboração). A razoabilidade por detrás desse método reside nas hipóteses de que indivíduos semelhantes possuem gostos semelhantes, ou de que existem padrões de compra associativas entre os itens – isto é, "quem compra o item x, geralmente também compra o item y". Assim, não há uso das informações dos usuários e dos itens, com exceção de uma matriz de avaliações parcialmente observada.

3. **Recomendações baseadas em conhecimento:** é um tipo específico de sistema de recomendação que se baseia no conhecimento explícito sobre a variedade de itens, preferências do usuário e critérios de recomendação. São empregados em cenários onde as demais abordagens não podem ser aplicadas.

Embora os métodos baseados em conteúdo e de FC possam ser utilizados de modo autossuficiente em uma implementação *end-to-end*, é comum também a utilização de abordagens híbridas que combinam duas técnicas em menor ou maior grau [Burke, 2002; Tran and Cohen, 2000; Burke, 2007; Gunawardana and Meek, 2009].

No domínio dos sistemas de recomendação, a FC é o método mais bem-sucedido [Sarwar et al., 2001, 2000], sendo largamente utilizada nos mais variados contextos. A filtragem colaborativa pode ser implementada à partir de duas estratégias principais, *(i)* baseada em memória, e *(ii)* baseada em modelo.

II.1.1 Filtragem Colaborativa Baseada em Memória

A FC baseada em memória utiliza todos os dados do item do usuário para gerar previsões. O sistema utiliza métodos estatísticos para buscar um conjunto de usuários que possuam histórico de transações semelhantes ao usuário ativo. Este método também é chamado de vizinho mais próximo ou FC baseada no usuário [Sarwar et al., 2001]. Bobadilla et al. [2013] explicam que existem três processos no método do vizinho mais próximo:

1. Escolher outros usuários semelhantes a um usuário específico;

2. Prever a classificação do item $i_j \in I$ para um usuário u_k qualquer, calculando os resultados da agregação de usuários semelhantes; e

3. Fornecer recomendações com base nos resultados previstos no estágio 2.

Su and Khoshgoftaar [2009] afirmam que as vantagens da FC baseada em memória são a facilidade de implementação e a alta capacidade que essas abordagens têm de acomodar os novos dados. No entanto, ela tem desempenho decrescente em dados com alta esparsidade, além de escalabilidade limitada para grandes conjuntos de dados [Su and Khoshgoftaar, 2009].

II.1.2 Filtragem Colaborativa Baseada em Modelo

A FC baseada em modelo fornece recomendações desenvolvendo um modelo a partir das avaliações dos usuários [Sarwar et al., 2001]. Para desenvolver um modelo, existem duas abordagens que podem ser usadas, que são abordagem de probabilidade ou previsão de classificação [Sarwar et al., 2001]. O processo de modelagem é conduzido por técnicas de aprendizado de máquina, como classificação, agrupamento e abordagens baseadas em regras [Sarwar et al., 2001].

Su and Khoshgoftaar [2009] declaram que a abordagem baseada em modelo tem melhores previsões do que a baseada em memória. Ela também é capaz de lidar com o problema de esparsidade e escalabilidade melhor do que algortimos baseados em memória. No entanto, a abordagem baseada em modelo requer maiores recursos para desenvolvimento dos modelos, como tempo e memória. Além disso, ela pode perder informações ao usar técnicas de redução de dimensionalidade [Su and Khoshgoftaar, 2009].

II.2 Grafos e Estruturas Comunitárias

Um grafo (ou rede) consiste, basicamente, de um conjunto de elementos – chamados nós ou vértices – com as conexões entre eles chamadas de arestas [Biggs et al., 1986]. Essas conexões podem ser representadas de várias maneiras: são ponderadas quando há atribuição de pesos específicos a cada aresta, e não ponderadas quando não há atribuição de pesos nas relações entre os nós. Além disso, as arestas podem ser direcionadas, indicando uma relação unidirecional entre os nós, ou não direcionadas, onde não há uma orientação definida nas conexões entre eles. A representação matricial dessas conexões é conhecida como matriz de adjacência A, na qual cada entrada a_{ij} é igual a 1 se existe uma aresta entre os nós i e j, e 0 caso contrário[1].

Essa estrutura é empregada no estudo das relações entre os objetos de um determinado conjunto. Sistemas que podem ser representados por grafos estão em toda parte, e muitos problemas de interesse prático podem ser modelados através dessa estrutura. A Figura II.3 apresenta um exemplo de um grafo fictício não ponderado e não direcionado, com seus nós em azul e suas arestas dispostas em cinza. A matriz de adjacência dessa rede é ilustrada na Figura II.4.

Pesquisas envolvendo associações no contexto de sistemas de recomendação são geralmente im-

[1]No caso de arestas ponderadas, a matriz de adjacência é preenchida com os pesos das associações, ao invés de valores binários

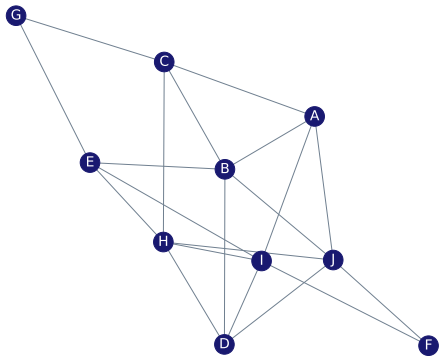


Figura II.3: Exemplo de um grafo não ponderado e não direcionado.

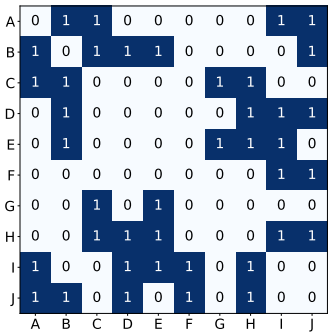


Figura II.4: Matriz de adjacência do grafo ilustrado na Figura II.3.

plementadas utilizando-se modelos baseados em grafos por duas razões. Primeiro, um grafo, ou modelo baseado em rede, é facilmente interpretado, e fornece uma estrutura mais natural e intuitiva para esse tipo aplicação. Em segundo lugar, muitos dos algoritmos baseados em grafos podem ser implementados diretamente em vários domínios comuns aos sistemas de recomendação, incluindo *e-commerce* e consumo de conteúdos digitais [Putra et al., 2017].

De modo prático, a visualização dos dados transacionais pode ser expressa com o usuário e o item sendo representados pelos vértices, e a transação como arestas representando tal relacionamento. Estruturas deste tipo – onde há dois tipos distintos de vértices (elementos) sendo que só há relacionamento entre os vértices de tipos diferentes, podem ser representados por grafos bipartidos, conforme ilustrado pela Figura II.5.

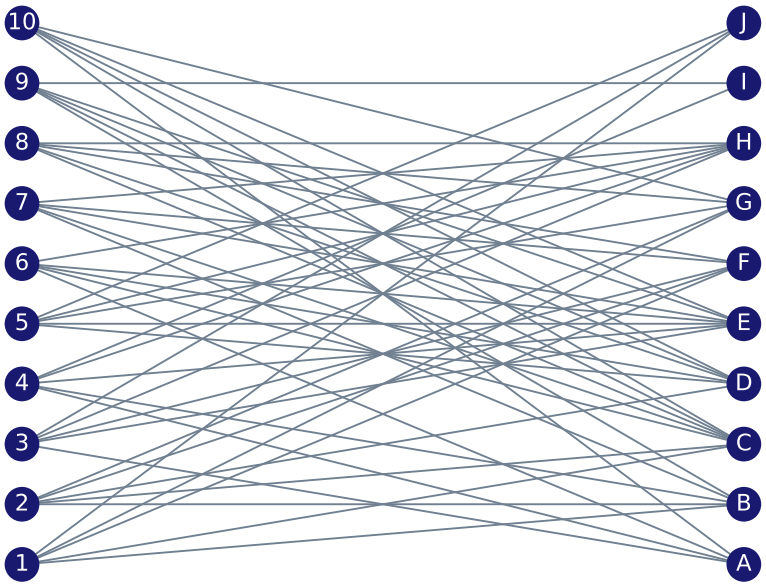


Figura II.5: Exemplos de um grafo bipartido com dois grupos de nós.

Esta representação, também conhecida pelos termos “redes bipartidas”, “ bigrafos” e “redes de

afiliação" [Lattanzi and Sivakumar, 2009; Borgatti and Halgin, 2011] é um tipo de grafo onde o conjunto de vértices pode ser dividido em dois conjuntos disjuntos de vértices, de modo que todas as arestas do grafo conectam apenas vértices de um conjunto à vértices do outro conjunto, mas nunca vértices que pertençam ao mesmo conjunto [West et al., 2001; Zhang and Chartrand, 2006]. Essa propriedade torna os grafos bipartidos especialmente úteis para modelar uma grande gama de relacionamentos, incluindo aquelas observadas em sistemas peculiares de recomendações [Manning et al., 2008; Zhang and Liu, 2010; Newman, 2010; Leskovec et al., 2014; Ricci et al., 2015]. No entanto, uma vez que a maioria das pesquisas teóricas se concentra na topologia de grafos unipartidos, estudos empíricos de sistemas de recomendação baseados em grafos bipartidos costumam re-esquematizar o problema em dois grafos unipartidos, com cada grafo possuindo apenas um tipo de nó que está sendo analisado em relação às propriedades topológicas. Zhou et al. [2007] fornecem orientações sobre como efetuar uma projeção de grafos bipartidos em grafos unipartidos.

II.2.1 Detecção de Comunidades

Um tópico de relevante interesse associado à redes é o da detecção ou caracterização de comunidades, que podem ser compreendidas como grupos de vértices densamente conectados e que, para além disso, parecem possuir certa independência do restante do grafo, com o qual compartilham apenas algumas arestas esparsas, enquanto que as relações entre os membros do mesmo grupo parecem ser nitidamente mais fortes [Tang and Liu, 2010; Newman and Girvan, 2004; Danon et al., 2005; Newman, 2006a; Fortunato and Castellano, 2007]. Esses grupos de vértices, ou comunidades, podem ser considerados compartimentos independentes [Fortunato and Castellano, 2007]. Sob a ótica de recomendação, os grupos mostrados na Figura II.6 podem representar, por exemplo, *clusters* de usuários que compartilham gostos semelhantes.

Dada uma rede $G = (V, E, W)$ onde V representa o conjunto de nós, E o conjunto de arestas e W o conjunto de pesos associados a cada aresta do grafo, o problema de detecção – ou clusterização – de comunidades pode ser formalmente definido como encontrar a partição $C = C_1 \cup \cdots \cup C_k$, que fornece as melhores comunidades em algum sentido [Zhao et al., 2011]. Os conjuntos de nós – ou comunidades – $C_1, \cdots, C_k$ são geralmente considerados disjuntos, embora existam trabalhos sobre a detecção de comunidades sobrepostas [Palla et al., 2005; Airoldi et al., 2008]

Diferentes métricas podem ser utilizadas para verificar a qualidade das divisões efetuadas, como índice de similaridade, conectividade, densidade inter e intra-*cluster*, etc [Choudhary et al., 2023]. A mais utilizada dessas métricas, no entanto, é a modularidade, introduzida por Newman and Girvan [2004] e ilustrada na Equação (II.2) [Newman, 2018].

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{i,j} - \lambda \frac{d_i d_j}{2m} \right) \delta(C_i, C_j) \tag{II.2}$$

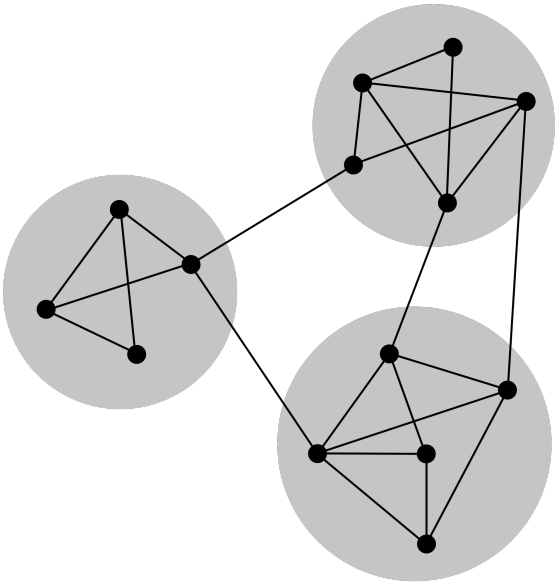


Figura II.6: Exemplo de uma rede com três estruturas comunitárias. As conexões dentro de cada comunidade são densas e robustas, enquanto que os pontos de contato (arestas) intercomunitários são esparsos.

onde m é a soma de todos os pesos das arestas [Blondel et al., 2008a], A é a matriz de adjacência da rede, d_i e d_j são respectivamente os graus ponderados dos nós i e j [Abbasi and Altmann, 2011], λ é o parâmetro de resolução e $\delta(C_i, C_j)$ é a função delta de Kronecker que assume o valor de 1 se i e j forem da mesma comunidade, e 0 caso contrário.

Nesse contexto, o componente $(A_{i,j} - \lambda\frac{d_i dj}{2m})$ mede se a interação entre os vértices i e j é mais intensa ou menos intensa do que seria esperado em uma rede aleatória equivalente. Uma diferença positiva indica que a interação entre esses nós é mais forte do que o esperado, enquanto uma diferença negativa sugere uma interação mais fraca do que o esperado. O parâmetro de resolução, por sua vez, estabelece um equilíbrio arbitrário entre arestas intra-grupo e arestas inter-grupo. Padrões de agrupamento mais complexos podem ser descobertos ao analisar a mesma rede com múltiplos valores de λ e, posteriormente, combinar os resultados [Reichardt and Bornholdt, 2006]. Embora seja comum utilizar $\lambda = 1$, Newman [2016] fornece um amplo estudo sobre a otimização desse parâmetro. Para cálculo da modularidade, somam-se os desvios da aleatoriedade para todos os pares de vértices que estão na mesma comunidade. Na Equação (II.2), $\delta(C_i, C_j)$, a função delta de Kronecker, limita a soma aos pares de vértices de mesmo grupo ($C_i = C_j$). Por fim, o coeficiente $\frac{1}{2m}$ normaliza a modularidade para o intervalo $[-1, 1]$ [Ganji et al., 2015], onde altos valores de Q retraram uma maior definição da estrutura modular da rede, ou seja, mais robustez ou qualidade do particionamento executado.

Em se tratando especificamente dos detectores de comunidade, já existe uma grande variedade de algoritmos na literatura [Choudhary et al., 2023], os quais podem ser classificados como tradicionais ou baseados em modularidade [Javed et al., 2018]. Algoritmos tradicionais são fundamentados

em várias categorias de técnicas de clusterização, tais como *clustering* particional [MacQueen, 1967; Dunn, 1973; Lloyd, 1982], *clustering* hierárquico [Girvan and Newman, 2001; Newman, 2004, 2006b; Morvan et al., 2017], *clustering* espectral [Barnard et al., 1995; Ding, 2004] e particionamento de grafos [Kernighan and Lin, 1970; Pothen, 1997]. A principal desvantagem desses algoritmos é o alto processamento computacional, e alguns deles podem funcionar apenas para comunidades disjuntas [Javed et al., 2018; Bourhim et al., 2018a].

Algoritmos baseados em modularidade, por sua vez, maximizam a modularidade. Encontrar o melhor valor desta medida é um problema extensivamente estudado, e vários algoritmos foram propostos, como algoritmos de otimização de modularidade [Blondel et al., 2008b; Newman, 2004], algoritmos genéticos [Tasgin et al., 2007; Zeng and Liu, 2015] e *simulated annealing* [Guimera and Amaral, 2005]. Comparados aos algoritmos tradicionais, as abordagens baseadas em modularidade tendem a gerar partições mais robustas, além de apresentarem um ganho significativo em termos de esforço computacional. Por conta desses fatores, esses algoritmos são muito utilizados em trabalhos que associam agrupamento de comunidades e sistemas de recomendação em algum grau, com destaque para o algoritmo de Louvain [Blondel et al., 2008c].

Algoritmo de Louvain

O método de Louvain para detecção de comunidades não sobrepostas em grandes redes é amplamente reconhecido e utilizado na literatura. Essa abordagem foi desenvolvida por Blondel et al. [2008c] à partir do trabalho anteriormente realizado por Girvan and Newman [2002]. Os autores eram da Universidade de Louvain, que emprestou seu nome à abordagem. O algoritmo é um método de otimização ganancioso que busca maximizar a modularidade e opera com uma complexidade de $O(n \cdot \log n)$, onde n representa o número de nós na rede [Lancichinetti and Fortunato, 2009].

O algoritmo funciona em duas etapas. Na primeira delas, ele atribui a cada nó sua própria comunidade e, em seguida, para cada nó, tenta encontrar o ganho de modularidade máximo movendo cada vértice para todas as comunidades vizinhas. Se nenhum ganho positivo for alcançado, o nó permanece em sua comunidade original. Esse processo continua até que nenhum movimento individual possa melhorar a modularidade.

O ganho de modularidade obtido ao mover um nó isolado i para uma comunidade C_j, onde $i \neq j$, pode ser facilmente calculado pela Equação (II.3) [Blondel et al., 2008a; Traag et al., 2019].

$$\Delta Q = \frac{d_{i,\text{in}}}{2m} - \gamma \frac{\Sigma_{\text{tot}} \cdot d_i}{2m^2} \tag{II.3}$$

onde m é a soma dos pesos de todos as arestas da rede ponderada, $d_{i,\text{in}}$ é a soma das pesos das arestas que conectam i aos nós em C_j, d_i é a soma dos pesos das arestas incidentes no nó i, Σ_{tot} é a soma dos pesos das arestas incidentes nos nós em C_j, e γ é o parâmetro de resolução.

No caso direcionado, o ganho de modularidade pode ser calculado usando a fórmula disposta na Equação (II.4) [Dugué and Perez, 2015].

$$\Delta Q = \frac{d_{i,\text{in}}}{m} - \gamma \frac{d_i^{\text{out}} \cdot \Sigma_{\text{tot}}^{\text{in}} + d_i^{\text{in}} \cdot \Sigma_{\text{tot}}^{\text{out}}}{m^2} \tag{II.4}$$

onde d_i^{out} e d_i^{in} são os graus ponderados externos e internos do nó i, e $\sum_{tot}^{in}$ e $\sum_{tot}^{out}$ são a soma das arestas de entrada e saída incidentes nos nós em C_j.

A segunda fase, por sua vez, consiste em construir uma nova rede cujos nós são agora as comunidades encontradas na primeira fase. Para realizar esta tarefa, os pesos das arestas entre os novos nós são dados pela soma do peso das arestas entre nós nas duas comunidades correspondentes. Uma vez que esta fase está completa, é possível reaplicar a primeira fase, criando comunidades maiores com modularidade aumentada. Todo esse processo é executado até que nenhum ganho de modularidade seja alcançado, ou que ao menos o valor atingido seja menor que algum limiar estabelecido.

A Figura II.7 ilustra esse processo, onde primeiro a modularidade é otimizada, permitindo apenas mudanças locais nas comunidades; na sequência, as comunidades encontradas são agregadas para construir uma nova rede de comunidades. Os passos são repetidos iterativamente até que não seja mais possível aumentar a modularidade. Os valores constantes na imagem à partir da fase de agregação representam as somas dos pesos das arestas intra-grupo e inter-grupo, oferecendo uma visualização das interações entre as comunidades identificadas.

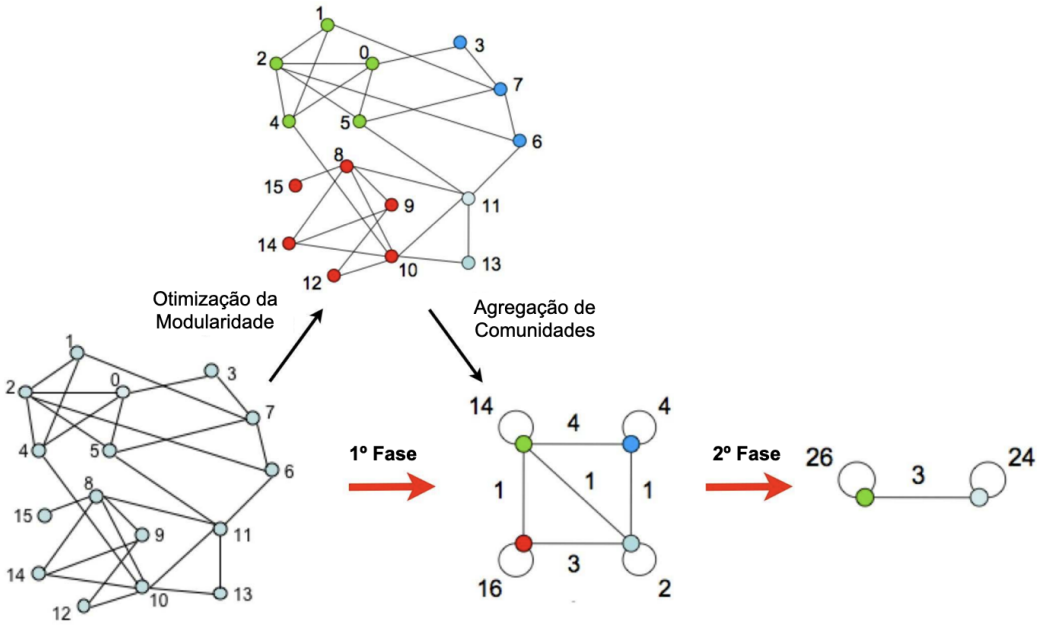


Figura II.7: Visualização das etapas do algoritmo de Louvain. Adaptado de Blondel et al. [2008a]

Algoritmo de Paris

Apesar de sua eficiência na produção de agrupamentos significativos, o algoritmo de Louvain não permite uma análise do grafo em diferentes escalas [Fortunato and Barthelemy, 2007; Lancichinetti and Fortunato, 2011]. Além disso, o parâmetro de resolução disponível nas versões preliminares do algoritmo não está diretamente relacionado ao número ou ao tamanho dos *clusters* e, portanto, é difícil de ajustar na prática. Para resolver essa questão, o algoritmo de Paris [Bonald et al., 2018] foi proposto como uma abordagem hierárquica inovadora para o agrupamento de grafos, almejando superar as limitações observadas no Louvain. O principal aperfeiçoamento reside na introdução de uma nova noção de distância entre *clusters*, derivada da probabilidade de amostragem de pares de nós, que é reduzível. Isso garante a representação eficiente da hierarquia resultante por dendrogramas regulares [Murtagh and Contreras, 2012].

A distância entre dois clusters a e b, P_{ij} pode ser expressa pela Equação (II.5), onde P_{ij} representa o peso da aresta entre os nós i e j. Essa distância é essencial para a fusão iterativa de *clusters* [Bonald et al., 2018].

$$d(a,b) = \sum_{i \in a, j \in b} P_{ij} \tag{II.5}$$

O algoritmo de Paris inicia com a maximização da resolução, onde cada nó é um *cluster* próprio. A fusão de *clusters* é realizada de forma iterativa, identificando os dois agrupamentos mais próximos em termos da nova distância introduzida e agregando-os. A sequência de fusões gera uma hierarquia completa e a lista resultante é devolvida como saída [Murtagh and Contreras, 2012].

Para além disso, o método utiliza o esquema de corrente de vizinhos mais próximos para garantir uma implementação eficiente, reduzindo a complexidade para algo próximo de $O(m)$, onde m é o número de arestas do grafo. Esta abordagem difere de Louvain, proporcionando uma visão hierárquica completa das estruturas presentes nos dados de rede. No entanto, a modularidade também é utilizada como métrica padrão para avaliar a qualidade do agrupamento gerado.

Capítulo III Trabalhos Relacionados

Uma atenção significativa tem sido dada nos últimos anos à sistemas de recomendações baseados em comunidades ou *clusters*. Essas abordagens visam melhorar a precisão e diversidade das abordagens clássicas de recomendações à partir de um maior aproveitamento dos relacionamentos presentes nos dados interacionais. Esta seção fornece uma visão geral de alguns trabalhos relevantes que exploraram o tema, e foi construída à partir de uma revisão sistemática de artigos publicados em periódicos variados.

Para condução dessa pesquisa foram utilizadas *strings* de busca contendo palavras-chave com o objetivo de identificar artigos de referência e contribuições ao tema sob escopo. Refinamentos posteriores foram feitos para priorizar documentos que enfatizassem a comparação de métodos de recomendação com e sem detectores de comunidades. O objetivo desta revisão foi estabelecer uma base sólida para fundamentar conceitos importantes, compreender os tipos de trabalhos que estão sendo conduzidos e auxiliar na construção da dissertação. As subseções subsequentes exploram o processo de pesquisa e revisão desses trabalhos e, por fim, um resumo das obras mais pertinentes à pesquisa de dissertação em vigor.

III.1 Processo de Pesquisa

A base de dados *Scopus* foi utilizada para a pesquisa, a partir da seguinte *string* de busca: TITLE-ABS-KEY((“recommendation system” OR “recommender system” OR “collaborative filtering”) AND (“community detection” OR “community detector” OR “community based”)) AND (LIMIT-TO (DOCTYPE, “cp”) OR LIMIT-TO (DOCTYPE, “ar”)) AND (LIMIT-TO (LANGUAGE, “English”)) AND (LIMIT-TO (PUBYEAR, 2018-)) AND (LIMIT-TO (PUBSTAGE, “final”)). A busca retornou 170 artigos, para os quais alguns critérios de inclusão e exclusão foram aplicados após a leitura dos resumos. Os critérios de inclusão são apresentados na Tabela III.1.

Os critério de exclusão, por sua vez, foram definidos para filtrar melhor os resultados da busca, tendo como alvo manter apenas artigos que apresentassem uma abordagem, escopo ou metodologias que contribuíssem para esta pesquisa. A Tabela III.2 apresenta os critérios de exclusão considerados.

Após esse primeiro refinamento, cerca de 70% dos trabalhos foram eliminados. A exclusão desses trabalhos deu-se por razões como incompatibilidade dos métodos utilizados, alta semelhança à

Tabela III.1: Critérios de Inclusão

Código	Descrição
CI-1	Trabalhos recentes da área de Ciência da Computação e demais áreas afins já publicados e que envolvam aplicação de detectores de comunidade em sistemas de recomendação.
CI-2	Trabalhos recentes da área de Ciência da Computação e demais áreas afins já publicados e que envolvam comparação de algoritmos de recomendação com e sem detectores de comunidade.

Tabela III.2: Critérios de Exclusão

Código	Descrição
CE-1	O artigo é uma versão mais antiga de um outro já considerado.
CE-2	Não foi possível ter acesso ao trabalho completo.
CE-3	Publicações em que o conteúdo dispõe apenas de conceitos.
CE-4	Artigos que não satisfaçam aos critérios de inclusão.

trabalhos anteriores já selecionados, ou irrelevância para fins desta pesquisa de dissertação. Por fim, todos os artigos restantes foram separados para leitura completa. Pela necessidade de um material complementar de apoio, alguns outros artigos foram selecionados através da técnica de *Snowballing*. Na sequência são apresentados os 23 artigos que mais se relacionaram com o tema em estudo por incluírem no *pipeline* alguma etapa prévia de agrupamento para execução das recomendações à nível local.

III.2 Revisão da Literatura

Pham et al. [2011] propuseram uma abordagem de agrupamento utilizando informações sociais dos usuários para aprimorar a qualidade das recomendações. Seus algoritmos selecionam um subconjunto dos vizinhos mais próximos do usuário ativo e geram previsões ao aplicar uma agregação ponderada de suas avaliações. Ao prever a avaliação de um item para determinado usuário, é calculada uma soma ponderada usando todas as avaliações dos vizinhos para esse item. As avaliações feitas com dados da *Digital Bibliographic Library Browser (DBLP)*[1] e do Epinion[2] evidenciaram que a técnica de agrupamento proposta superou os algoritmos tradicionais de filtragem colaborativa em termos de desempenho das recomendações.

A introdução de uma abordagem de FC baseada em comunidades para sistemas de recomendação foi introduzida por Gorripati and Vatsavayi [2017] para abordar os problemas de esparsidade de dados e *cold-start*. Na abordagem proposta pelos autores foram exploradas as preferências dos

[1]Disponível em `https://snap.stanford.edu/data/com-DBLP.html`.
[2]Disponível em `https://snap.stanford.edu/data/soc-Epinions1.html`.

usuários dentro da mesma comunidade. O método proposto envolve identificar padrões dos usuários com base nos itens que eles avaliaram, construir uma rede usando relações de coavaliação, identificar estruturas de comunidade por meio de um algoritmo de propagação de rótulo e calcular a similaridade entre vizinhos com base em usuários altamente correlacionados e confiáveis. Avaliações experimentais no conjunto de dados Movielens[3] demonstraram uma melhoria na qualidade das recomendações em comparação com algoritmos tradicionais.

Bourhim et al. [2018b, 2019] propuseram uma abordagem de FC baseada em grafo para sistemas de recomendação. O método sugerido em Bourhim et al. [2018b] concentra-se em refinar a homofilia no nível da comunidade, tratando os usuários como uma rede de subcomunidades altamente semelhantes. Ele introduz um algoritmo que aproveita o conceito de nós-chave para melhorar a precisão das previsões em comparação com os métodos tradicionais baseados apenas na similaridade entre usuários. O método proposto utiliza algoritmos baseados em modularidade de Louvain para agrupar usuários. Ao identificar nós-chave dentro das comunidades identificadas, os autores obtiveram ganhos significativos de desempenho em comparação às abordagens *Similarity-Based Model (SIM)*, um método de FC baseado em memória, e *Singular Value Decomposition (SVD)* – um método de FC baseado em modelo.

Em Bourhim et al. [2019], o estudo apresentado avaliou a abordagem proposta em [Bourhim et al., 2018b] experimentando uma variedade de outros métodos de detecção de comunidade para aprimorar a precisão e o tempo de execução do algoritmo. Por meio de testes nos conjuntos de dados Movielens e YOW[4], observou-se que os algoritmos *Agglomerative Hierarchical Clustering (AHC)* e *k-means* exibiram maior utilidade do que o algoritmo Louvain ao lidar com grandes conjuntos de dados.

Uma evolução da arquitetura desenvolvida por Bourhim et al. [2018b] foi posteriormente proposta por Jain et al. [2021], onde as recomendações de músicas são feitas com base nas escolhas feitas por especialistas pertencentes às comunidades. No trabalho desenvolvido pelos autores, agrupamentos de usuários são determinados utilizando-se o método Louvain. Em seguida, indivíduos especialistas são mapeados em cada uma das comunidades identificadas com base no grau de centralidade dos nós. As escolhas feitas por esses indivíduos são utilizadas como recomendações para os demais usuários do mesmo agrupamento. Avaliações realizadas em um conjunto de dados coletados da Deezer[5] demonstraram bom desempenho da abordagem proposta.

Cao et al. [2020] apresentaram uma interessante proposição baseada em detecção de comunidades para os desafios de esparsidade de dados e *cold-start* em sistemas de recomendação de músicas. Na pesquisa desenvolvida, as iterações entre usuários e músicas são inicialmente estruturadas como uma

[3]Disponível em `https://movielens.org/`.
[4]Disponível em `https://users.soe.ucsc.edu/~yiz/papers/data/YOWStudy/`.
[5]`https://www.deezer.com/`

rede bipartida. Projeções unipartidadas dessa rede são realizada através da utilização do coeficiente de correlação de Pearson. Na sequência, o método de Louvain é utilizado para detectar comunidades de usuários e músicas nas projeções construídas. Os resultados experimentais nos conjuntos de dados Million Song[6], Yahoo Music[7] e Movielens demonstraram que o algoritmo proposto obteve maior precisão, estabilidade e menor tempo de execução em comparação tanto ao algoritmo SVD quanto em relação à um método de *cluster* bipartido, mostrando uma melhoria na precisão das recomendações e um desempenho consistente.

Çiftçi et al. [2021], por sua vez, propuseram uma abordagem baseada em grafo que utiliza regras de associação extraídas das *playlists* do Spotify[8] para identificação e recomendação de artistas similares. Os autores empregaram os algoritmos de detecção de comunidades de Louvain e de propagação de rótulo para identificar *clusters* de artistas relacionados. Os pesos das arestas na rede de artistas foram determinados com base em métricas de suporte e confiança, e os grupos detectados foram validados interna e externamente usando o *Adjusted Rand Index (ARI)* e o Coeficiente de Sobreposição em Pares. Através de uma análise experimental no conjunto de dados *Million Song* [Chen et al., 2018], os autores observaram altas concordâncias entre as clusterizações resultantes da abordagem proposta e as listas originais de artistas relacionados do Spotify.

Um novo método de recomendação que combina fatoração matricial, análise de redes e técnicas de aprendizado profundo para mitigação das problemáticas de esparcidade e *cold-start* foi proposto por Kherad and Bidgoly [2022]. Sua abordagem utiliza o algoritmo de Louvain para detecção de comunidades e o a técnica de HITS [Kleinberg, 1999; Langville and Meyer, 2005] para identificar nós importantes na rede. Resultados experimentais nos conjuntos de dados Ciao[9] e Epinions demonstraram a superioridade de seu método em comparação com os métodos de recomendação existentes.

Paleti et al. [2021] desenvolveram uma abordagem híbrida intitulada LA-ALS para lidar com o problema *cold-start* e fornecer recomendações eficazes. A abordagem utiliza o algoritmo de Louvain para analisar a relação entre usuários, e a técnica de mínimos quadrados alternados para prever recomendações. Assim, para um novo usuário qualquer, sua comunidade será identificada e as predições serão baseadas no modelo treinado para usuários semelhantes. Experimentos foram realizados utilizando conjuntos de dados do mundo real, como Movielens e bancos de dados do Facebook. A eficácia da abordagem LA-ALS foi demonstrada por meio de duas métricas, nomeadamente, *Mean Absolut Error (MAE)* e *RMSE*. Os resultados mostraram que a abordagem LA-ALS gerou recomendações melhores quando comparada com técnicas existentes, como k-NN e SVD.

Zhang et al. [2017] propuseram um sistema de recomendação de músicas que considera conjuntamente as preferências de longo e curto prazo em sua arquitetura. Para representar as preferências

[6]Disponível em `http://millionsongdataset.com/`.
[7]Disponível em `https://webscope.sandbox.yahoo.com/catalog.php?datatype=r`.
[8]`https://open.spotify.com/`
[9]Disponível em `https://www.cse.msu.edu/~tangjili/trust.html`.

de longo prazo, o estudo divide a rede de usuários em comunidades de acordo com padrões de comportamento semelhantes. Paralelamente, sugere-se fundir essa rede de comunidades com a cadeia de Markov para fazer recomendações sequenciais que também se adequem melhor às preferências de curto prazo. Para detecção das comunidades os autores avaliaram três algoritmos diferentes: *Walk trap* [Pons and Latapy, 2005], *Fast greedy* [Clauset et al., 2004] e Louvain. Resultados dos experimentos no *dataset* ChangBa (um aplicativo de música online na China com mais de 300 milhões de usuários) mostraram que a abordagem proposta superou os algoritmos existentes em termos da precisão na previsão das preferências de longo e curto prazo.

Xin et al. [2014], por sua vez, sugeriram um método de recomendações baseado em comunidades para recomendação de livros. Para fins de agrupamento dos usuários em comunidades com padrões semelhantes, os autores propuseram um novo algoritmo, chamado *Importance Greedy*. A ideia central da técnica proposta é selecionar os k nós mais importantes da rede como as comunidades iniciais. Um processo contínuo de inclusão dos demais nós às comunidades é conduzido até que não se possa mais observar aumento da modularidade. Os autores avaliam as métricas de peso médio, grau e intermediação para mensurar a importância que cada um dos nós possui na rede. Para recomendação de um livro qualquer à um usuário, utilizam-se as avaliações pretéritas dos nós vizinhos presentes nas comunidades identificadas. Ao implementar os métodos em um conjunto de dados composto pelo histórico de transações de uma biblioteca, observou-se que o método proposto apresentou desempenho superior aos algoritmos colaborativos tradicionais em termos de precisão e *recall*.

Para lidar com o desafios do *cold-start*, Pasricha and Solanki [2019] desenvolveram um sistema de recomendação com base na identificação de líderes de opinião em comunidades de redes sociais. Uma vez identificadas as comunidades, usuários-chave são mapeados em cada agrupamento à partir da centralidade de autovetor. Assim, em cada comunidade os itens são recomendados com base nos interesses e preferências do respectivo líder de opinião. Os autores realizaram análises em dados reais coletados ao longo de dois meses na plataforma Twitter. Embora os autores tenham reforçado que algoritmos variados possam ser usados na etapa de agrupamento dos usuários, para fins experimentais eles se valeram de uma técnica de mineração de dados com base em atributos útei para consecução do agrupamento, como localização do usuário, idioma e interesses baseados em palavras-chave que foram obtidos durante a coleta dos dados.

Sharma and Bedi [2017] construíram um Sistema de Recomendação Colaborativo Baseado em Comunidade (CCFRS, do inglês *Community based Collaborative Filtering Recommender System*). O método de Louvain para detecção de comunidades é aplicado para descobrir comunidades nos conjuntos de dados. O método de geração de recomendações é baseado na ideia proposta do escore de Frequência de Item - Frequência Inversa da Comunidade (IF-ICF) de cada item na comunidade

do usuário-alvo. Os escores IF ajudam a encontrar o conjunto de itens que são exclusivos para uma comunidade específica. Os valores ICF são inversamente proporcionais ao número de comunidades nas quais um item foi avaliado. Os escores IF-ICF dos itens são posteriormente utilizados para encontrar os escores de previsão dos itens não vistos pelo usuário, a fim de apresentar um conjunto das "n" melhores recomendações ao usuário. Um protótipo do sistema foi desenvolvido usando Java, e análises experimentais foram realizadas para o domínio de livros utilizando o conjunto de dados Book-Crossing[10].

Hadil and Mandi [2022] apresentaram informações demográficas de usuários em nível comunitário como uma nova dimensão para recomendações. O método proposto pelos autores utiliza o algoritmo de Louvain para descobrir comunidades de usuários à partir da similaridade demográfica usuário-usuário. Uma técnica de FC com base na matriz de item de usuário é usado para gerar previsão de classificação. A abordagem melhorou o problema de escalabilidade, *cold-start* e escassez de dados do sistema de recomendação baseado em FC. Experimentos foram realizados no conjunto de dados Movielens para prever a classificação de filmes e produzir as melhores recomendações para novos usuários, e os resultados foram comparados à nível de RMSE com um sistema de FC baseado no usuário.

Uma análise sobre a aplicação de algoritmos de particionamento em registros de classificação para FC foi conduzida por O'Connor and Herlocker [1999]. O objetivo primário dos autores foi avaliar se o agrupamento dos itens com base em classificações de usuários pode melhorar previsões e escalabilidade. Quatro conhecidos algoritmos de agrupamento foram experimentados, AHC, *Robust Clustering using Links (ROCK)*, *k-Way Multilevel Partitioning (KMetis)* e *h-Way Multilevel Partitioning (HMetis)*; e os particionamentos gerados por essas abordagens foram comparados ao particionamento de gênero e também à um agrupamento aleatório. Resultados obtidos no conjunto de dados Movielens-100k demonstraram que o particionamento proposto pode aumentar a escalabilidade, mas a melhoria na precisão das previsões é variável. O algoritmo KMetis mostrou-se promissor, proporcionando previsões mais precisas do que o particionamento aleatório, e semelhantes às obtidas com base em gêneros.

Sarwar et al. [2002] apresentaram e avaliaram experimentalmente uma nova abordagem para melhorar a escalabilidade de sistemas de recomendação usando técnicas de agrupamento. Experimentos conduzidos no conjunto de dados Movielens-100k mostraram que o agrupamento baseado em vizinhos – com destaque para a técnica de *bisecting k-means* – proporcionou uma qualidade de previsão comparável à abordagem padrão de FC em termos de MAE e, ao mesmo tempo, melhorou significativamente o desempenho *online* do sistema.

Quan et al. [2006] propuseram um método de agrupamento de itens, de modo que dentro de um

[10]Disponível em `https://grouplens.org/datasets/book-crossing/`.

cluster a similaridade entre usuários não mude significativamente. Assim, ao prever a avaliação de um usuário para um item, os autores agregam apenas as avaliações de usuários que têm um alto grau de similaridade com aquele usuário dentro do *cluster* ao qual o item pertence. Experimentos avaliando a abordagem foram realizados em um conjunto de dados reais retirado do sistema de recomendação de filmes do site Movielens. Resultados preliminares com base no MAE demonstraram que a abordagem foi capaz de melhorar a precisão das previsões de recomendações em comparação com abordagens existentes

Ma et al. [2014a] examinaram os relacionamentos de confiança para destacar usuários considerados especialistas na comunidade-alvo. No caso do *cold-start*, o novo usuário é automaticamente conectado aos especialistas identificados por meio de um relacionamento de confiança. A justificativa é que existe uma correlação entre confiança e similaridade de usuários [Ma et al., 2014b], de modo que as avaliações ausentes do usuário em determinados itens são estimadas pela média ponderada das avaliações de seus vizinhos confiáveis que avaliaram os mesmos itens na comunidade do usuário. Os autores utilizaram o algoritmo de SVD na matriz de relacionamento de confiança para detectar comunidades de confiança. Uma estratégia de caminhada aleatória (*random walk*) foi utilizada para explorar cada comunidade e coletar os vizinhos confiáveis para os usuários. Então, as previsões puderam ser geradas pela FC tradicional com base nos perfis de classificação do usuário complementados. Experimentos com o conjunto de dados Epinions demonstraram que a abordagem proposta superou as abordagens de comparação tanto em RMSE quanto em cobertura.

Lalwani et al. [2015] introduziram as interações sociais por comunidade como uma nova dimensão para recomendações e apresentaram um sistema de recomendação social utilizando abordagens de FC e detecção de comunidades. Os autores utilizam (i) algoritmo de detecção de comunidades de Louvain para extrair relações de amizade entre usuários, analisando a rede social usuário-usuário e (ii) FC baseada em usuário-item para previsão de classificação. Eles desenvolveram a abordagem utilizando o framework *map-reduce*. O método melhorou a escalabilidade, cobertura e o problema de *cold-start* do sistema de recomendação baseado em FC. Foram realizados experimentos nos conjuntos de dados do Movielens e do Facebook para prever a classificação do filme e produzir recomendações dos *top-k* para novos usuários

Uma notação matricial para capturar o nível de pertencimento que cada usuário teria às comunidades existentes foi introduzida por Sahebi and Cohen [2011]. A matriz fornece evidências para avaliar a similaridade entre os usuários de acordo com as comunidades às quais eles pertencem. Os autores concentraram-se no problema de *cold-start* com a suposição de que, se o usuário é novo em um serviço online (por exemplo, Youtube), dados históricos de serviços distintos (por exemplo, Facebook) podem ser considerados para inicializar os algoritmos de recomendação. A avaliação experimental considerou múltiplos sinais, como conexões explícitas (ou seja, amizades), comentários

e avaliações; cada um explorado individualmente para definir comunidades. O conjunto de dados utilizado para fins experimentais foi o *imhonet*[11], uma rede social russa. Os autores demonstraram que realizar FC entre membros da comunidade é mais eficaz do que executar FC em todos os usuários. Além disso, explicitaram que o uso de outras dimensões de interesses do usuário ou conexões do usuário podem ajudar a obter um *Normalized Discounted Cumulative Gain (NDCG)* razoável no problema de *cold-start*.

Das et al. [2007] combinaram técnicas como FC, Mínima Função de Hashing (MinHash) [Broder, 1997] e *Probabilistic Latent Semantic Analysis (PLSA)* para agrupar usuários analisando o número de vezes que uma notícia foi visitada juntamente com outras histórias no histórico de cliques do usuário. Em particular, a MinHash é um método de agrupamento probabilístico que agrupa dois usuários com probabilidade proporcional à sobreposição entre o conjunto de notícias que esses usuários visitaram. O PLSA implementa a representação de baixa dimensão na qual usuários e histórias de notícias são mapeados. Os autores dedicaram muito esforço para a implementação de um algoritmo distribuído para garantir a escalabilidade da abordagem para o serviço *online* de notícias do Google. Experimentos foram conduzidos usando o Movielens, NewsSmall e NewsBig.

Xu et al. [2012] formularam o problema de *Multiclass Co-Clustering* para encontrar subgrupos significativos. Em seguida, os autores propuseram um framework unificado para estender os algoritmos tradicionais de FC, utilizando informações dos subgrupos para melhorar seu desempenho de recomendação. A abordagem pode ser vista como uma extensão dos modelos tradicionais de FC. Experimentos sistemáticos em três conjuntos de dados do mundo real (Amazon[12], Last.fm[13] e Movielens) demonstraram a eficácia da abordagem proposta.

Um sistema multiagente que imita a evolução de um grafo social e, especificamente, a criação e extinção das comunidades [Cazabet and Amblard, 2011] foi utilizado por Abdrabbah et al. [2016]. Eles modelaram as mudanças temporais nos interesses dos cidadãos nos serviços eletrônicos de governo para fins de recomendação. De fato, o mesmo serviço pode ser relevante em um determinado período, circunstância e se estiver relacionado a outros serviços, mas inútil de outra forma (por exemplo, uma certidão de nascimento necessária para a matrícula escolar). Um grafo de conhecimento representa essa informação por meio de arestas que conectam dois serviços se forem frequentemente usados por usuários em um determinado período de tempo. O grafo é submetido à detecção de comunidades e as preferências do usuário-alvo selecionam as comunidades mais próximas das quais os serviços candidatos são coletados e submetidos a uma FC tradicional baseada em itens. O conjunto de dados Movielens foi utilizado para avaliação experimental da abordagem proposta.

[11] `www.imhonet.ru`
[12] Disponível em `https://jmcauley.ucsd.edu/data/amazon/`.
[13] Disponível em `http://millionsongdataset.com/lastfm/`.

A Tabela III.3 resume os artigos selecionados pelos métodos utilizados para detecção (ou agrupamento) de comunidades, domínio de interesse e conjuntos de dados utilizados para fins experimentais.

#	**Referência**	**Domínio de Interesse**	**Dataset**	**Ténica de Agrupamento**
1	[O'Connor and Herlocker, 1999]	Filmes	Movielens	AHC, ROCK, KMetis e HMetis
2	[Sarwar et al., 2002]	Filmes	Movielens	*k-means*
3	[Quan et al., 2006]	Filmes	Movielens	Agrupamento baseado na estabilidade das similaridades dos usuários
4	[Das et al., 2007]	Filmes e Notícias	Movielens, NewsSmall e NewBig	PLSA
5	[Pham et al., 2011]	Interesse Acadêmico e *e-commerce*	DBLP e Epinion	Louvain
6	[Sahebi and Cohen, 2011]	Redes Sociais	imhonet	Louvain
7	[Xu et al., 2012]	*e-commerce*, Músicas e Filmes	Amazon, Last.fm e Movielens	*Multiclass Co-Clustering*
8	[Xin et al., 2014]	Livros	Próprio	*Importance Greedy*
9	[Ma et al., 2014a]	*e-commerce*	Epinions	SVD
10	[Lalwani et al., 2015]	Filmes e Redes Sociais	Movielens e Facebook	Louvain
11	[Abdrabbah et al., 2016]	Filmes	Movielens	*Multi-agent system*
12	[Gorripati and Vatsavayi, 2017]	Filmes	Movielens	Louvain
13	[Zhang et al., 2017]	Músicas	ChangBa	*Fast greedy*, *Walk trap* e Louvain
14	[Sharma and Bedi, 2017]	Livros	*Bookscrossing*	Louvain
15	[Bourhim et al., 2018b]	Filmes	Movielens	Louvain
16	[Bourhim et al., 2019]	Filmes e Notícias	Movielens e YOW	AHC, *k-means* e Louvain
17	[Pasricha and Solanki, 2019]	Redes Sociais	Twitter	Heurística
18	[Cao et al., 2020]	Filmes e Músicas	Million Song, Yahoo Music e Movielens	Louvain
19	[Jain et al., 2021]	Músicas	Deezer	Louvain
20	[Çiftçi et al., 2021]	Músicas	*Million Song*	Louvain

#	Referência	Domínio de Interesse	Dataset	Ténica de Agrupamento
21	[Paleti et al., 2021]	Filmes e Redes Sociais	Movielens e Facebook	Louvain
22	[Kherad and Bidgoly, 2022]	*e-commerce*	Ciao e Epinions	Louvain
23	[Hadil and Mandi, 2022]	Filmes	Movielens	Louvain

Tabela III.3: Lista de artigos da revisão sistemática com informações gerais resumidas.

Os trabalhos mencionados contribuíram para o campo ao mostrar o potencial da integração de algoritmos de detecção de comunidade (ou *clustering*) com sistemas de recomendação para aprimorar a precisão das recomendações e lidar com diversos desafios. No entanto, essas abordagens também possuem limitações. Por exemplo, alguns métodos focam principalmente em resolver o problema de *cold-start* ou melhorar a qualidade das recomendações em domínios específicos, enquanto outros podem ser limitados pela eficiência computacional ou pela escalabilidade de seus algoritmos. Além disso, existe uma lacuna significativa nas pesquisas atuais no que diz respeito à exploração de diferentes combinações de algoritmos de descoberta de comunidade e recomendação, e seu impacto no desempenho das previsões. Isso se deve, em grande parte, à dificuldade na comparação direta entre estudos anteriores, uma vez que eles variam substancialmente em métodos, conjuntos de dados e métricas de avaliação utilizados. Isso proporciona uma oportunidade para experimentação e investigação adicionais no campo, com uma ampla variedade de possibilidades a serem exploradas e comparadas.

Capítulo IV Abordagem Proposta

A motivação deste trabalho é explorar o potencial das técnicas de detecção de comunidade para aprimorar os algoritmos de recomendação. Essa abordagem oferece uma ampla aplicabilidade, podendo ser utilizada em diversos algoritmos de recomendações disponíveis, como aqueles baseados em heurísticas, métodos de vizinhança, fatorização de matriz e outras técnicas de filtragem colaborativa. O desenvolvimento dessa metodologia é composto por quatro etapas distintas que serão detalhadas adiante: *(i)* Projeção do Grafo Bipartido (Seção IV.1); *(ii)* Detecção de Comunidades (Seção IV.2); *(iii)* Instanciamento de Modelos e Treinamento (Seção IV.3); e *(iv)* Predição (Seção IV.4).

IV.1 Projeção do Grafo Bipartido

Nesta seção, delineamos o processo de transformação de um grafo bipartido ponderado, que representa avaliações usuários × itens, em um grafo ponderado unipartido focado exclusivamente nos usuários. Essa projeção nos permite capturar as relações e interações entre os usuários, considerando as conexões ponderadas derivadas de suas avaliações. A abordagem foi fortemente influenciada pelas sugestões e conceitos delineados no trabalho de Zhou et al. [2007], que forceneu um arcabouço teórico valioso sobre o tema.

Considere um grafo bipartido ponderado $\mathcal{G} = (U, I, E, r)$, onde U é o conjunto de vértices que retratam os usuários, I é o conjunto de vértices que representam os itens do catálogo, E é o conjunto de arestas que conectam vértices pertencentes ao conjunto U com vértices pertencentes ao conjunto I, e r representa as avaliações, ou seja, os pesos atribuídos a cada aresta. É possível avaliar a dissimilaridade entre dois usuários u_k e u_l, onde $(u_k, u_l) \in U^2$, percorrendo os itens que ambos os usuários avaliaram e calculando a diferença média ao quadrado entre suas avaliações correspondentes. Isso resultará em um grafo ponderado unipartido $\mathcal{G}' = (U, E', w)$, onde U' representa o conjunto de vértices na projeção ponderada – que corresponde ao conjunto de vértices U no grafo bipartido, E' representa o conjunto de arestas na projeção ponderada, capturando as conexões entre pares de vértices em U que avaliaram pelo menos um item em comum, e w representa a função de peso na projeção ponderada, atribuindo valores numéricos às arestas em E'.

Matematicamente, o peso w_{u_k,u_l} é definido como mostrado na Equação (IV.1):

$$w_{u_k,u_l} = \frac{1}{|\mathcal{G}[u_k] \cap \mathcal{G}[u_l]|} \left(\sum_{i \in \mathcal{G}[u_k] \cap \mathcal{G}[u_l]} (r_{u_k,i} - r_{u_l,i})^2 \right) \tag{IV.1}$$

onde $\mathcal{G}[u_k]$ e $\mathcal{G}[u_l]$ representam respectivamente os conjuntos de itens avaliados pelos usuários u_k e u_l. As variáveis $r_{u_k,i}$ e $r_{u_l,i}$, por sua vez, representam as avaliações do *i-ésimo* item feitas pelos usuários u_k e u_l, respectivamente. Para conferir uma conotação positiva aos pesos mencionados, aplicamos a transformação mostrada na Equação (IV.2), que também atua como uma normalização para o intervalo $(0, 1]$:

$$w'_{u_k,u_l} = \frac{1}{1 + w_{u_k,u_l}} \tag{IV.2}$$

Valores maiores de w resultarão em pesos normalizados w' menores, indicando uma influência ou similaridade menor entre o par de usuários. Por outro lado, valores menores de w levarão a pesos normalizados maiores, indicando uma influência ou similaridade maior. Assim, obtém-se as conexões ponderadas que refletem as relações entre os usuários no grafo bipartido. Essas conexões ponderadas desempenham um papel crucial na construção do grafo projetado, onde os usuários são representados como vértices, e os pesos das arestas são determinados pelos valores calculados de w'. Essa transformação permite focar nas relações entre usuários e estabelece a base para as etapas subsequentes de nossa metodologia de recomendação.

A Figura IV.1 fornece uma representação visual de um grafo bipartido $\mathcal{G} = (U, I, E, r)$. Este grafo foi gerado utilizando-se um conjunto de dados fictício composto por 100 avaliações atribuídas por 10 usuários (Usuário 1 a Usuário 10) a 10 itens diferentes (Item A a Item J), com avaliações variando de 1 a 5. Sua projeção correspondente $\mathcal{G}' = (U, E', w')$ é ilustrada na Figura IV.2, onde é possível ver as conexões entre os usuários com base em suas avaliações.

IV.2 Detecção de Comunidades

Nesta seção, discutimos o próximo passo de nossa metodologia, que envolve a detecção de comunidades no grafo projetado. A detecção de comunidades tem como objetivo identificar *clusters* de usuários no grafo que apresentam padrões ou comportamentos semelhantes. Ao descobrir essas comunidades, podemos obter informações sobre a estrutura subjacente e a dinâmica das interações entre usuários, o que pode aprimorar ainda mais a eficácia dos algoritmos de recomendação.

Para detectar comunidades no grafo projetado, podemos empregar vários algoritmos de descoberta de comunidades para analisar os padrões de conectividade e os pesos das arestas entre os usuários, a fim de identificar grupos densamente conectados. Algoritmos específicos têm sido amplamente utilizados para a detecção de comunidades em diversos campos de estudo, com destaque

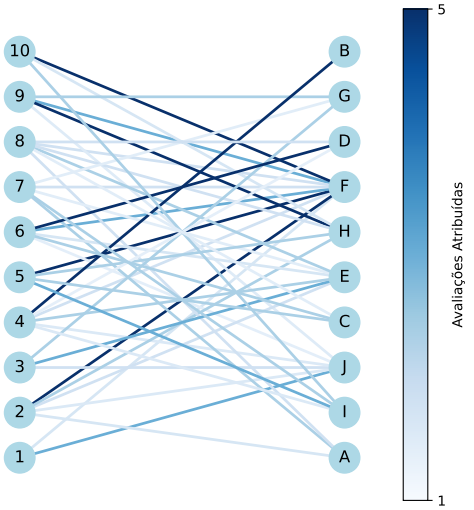


Figura IV.1: Exemplo de um grafo ponderado e bipartido e que representa as relações usuário-item .

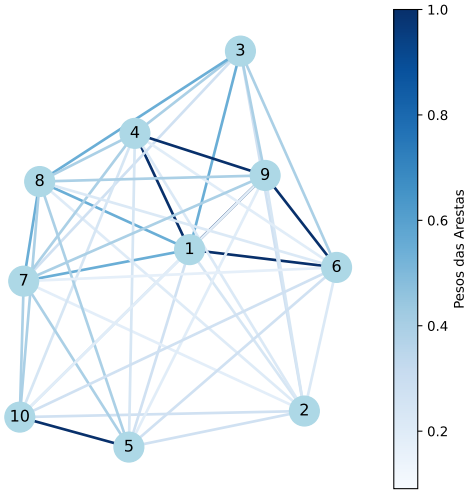


Figura IV.2: Projeção correspondente da rede ilustrada na Figura IV.1

para os algoritmos Louvain [Blondel et al., 2008c], Girvan-Newman [Girvan and Newman, 2002], Infomap [Rosvall and Bergstrom, 2008] e *Label Propagation* [Raghavan et al., 2007], devido à sua eficácia em capturar padrões e revelar agrupamentos significativos.

O processo de detecção de comunidades é ilustrado na Figura IV.3, onde o algoritmo de Louvain é aplicado ao grafo projetado gerado na subseção anterior (Figura IV.2). Essa aplicação resulta em uma visualização do grafo onde a cada nó é atribuída uma cor específica que representa sua pertinência a uma comunidade. Essa visualização ajuda a ilustrar a estrutura das comunidades e fornece uma compreensão clara dos grupos de usuários detectados.

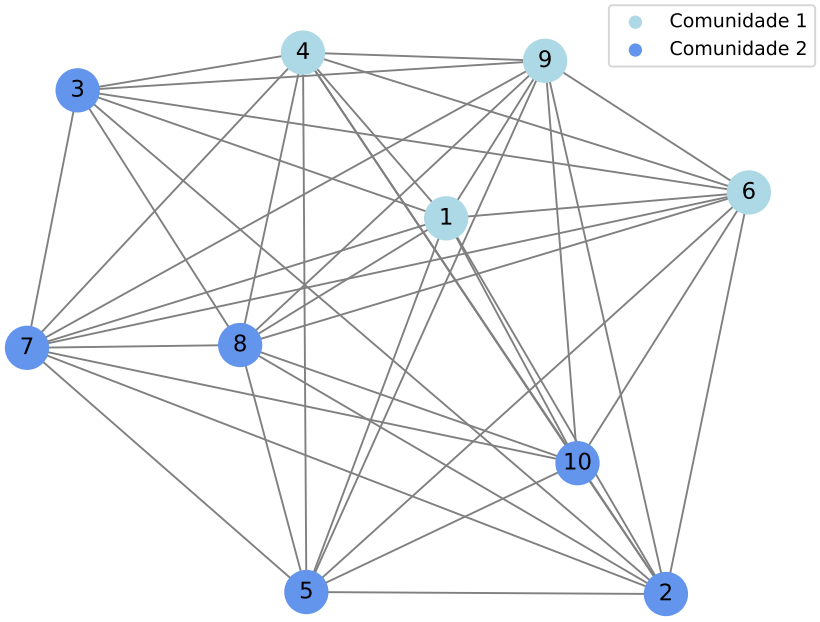


Figura IV.3: Projeção do grafo com duas comunidades identificadas.

Por meio das técnicas de detecção de comunidades, é possível obter informações valiosas sobre o comportamento coletivo dos usuários, permitindo adaptar os algoritmos de recomendação para atender melhor às preferências e necessidades específicas. Isso melhora a precisão e relevância das recomendações, levando, em última instância, a uma experiência do usuário mais personalizada e satisfatória. Uma vez que as comunidades são detectadas, cada comunidade funciona como um subconjunto de usuários com características ou preferências compartilhadas. Essas informações podem ser utilizadas para personalizar o processo de recomendação dentro de cada comunidade, levando em consideração as necessidades e preferências específicas de seus membros. Essa abordagem direcionada ajuda a fornecer recomendações mais precisas e relevantes aos usuários com base em suas afiliações comunitárias.

IV.3 Instanciamento de Modelos e Treinamento

Esta seção aborda a etapa de treinamento do modelo da metodologia de recomendação proposta neste trabalho. Após a conclusão das etapas de projeção do grafo e detecção de comunidades, as informações sobre as comunidades identificadas podem ser utilizadas para aprimorar o processo de treinamento de um modelo de recomendação. O objetivo é incorporar as informações das comunidades para capturar o comportamento e as preferências coletivas dos usuários dentro de cada comunidade.

Vamos denotar o conjunto de dados de treinamento completo como D, um algoritmo específico de recomendação como A e o conjunto de comunidades como C. O processo para o treinamento do modelo de recomendação baseado em comunidades pode ser descrito da seguinte forma.

Considere que uma amostra do conjunto de dados de treinamento D_i seja preparada especificamente para a comunidade C_i. Este subconjunto de dados consiste nas avaliações e preferências dos usuários na comunidade $C_i \in C$. Utilizando essa porção do conjunto de dados completo D, um modelo de recomendação M_i, que está associado à comunidade C_i, pode ser treinado utilizando o algoritmo de recomendação A. O modelo M_i aprende a partir das interações usuário-item dentro de D_i, capturando as preferências e padrões específicos da comunidade C_i.

Após o treinamento do modelo de recomendação M_i, ele é armazenado como o modelo treinado específico da comunidade C_i. Todo esse processo é repetido para cada comunidade $C_i \in C$, resultando em um conjunto de modelos de recomendação treinados $M = \{M_1, M_2, \cdots, M_{|C|}\}$ para cada uma das $|C|$ comunidades. O Algoritmo 1 resume as etapas envolvidas no treinamento do modelo de recomendação baseado em comunidades.

Como exemplo ilustrativo, no caso do grafo projetado fictício com duas comunidades (Figura IV.3), esse processo de treinamento iterativo teria gerado duas instâncias de treinamento. Uma instância incluiria os usuários pertencentes à primeira comunidade, nomeadamente usuários 1,

Algorithm 1 : Treinamento Baseado em Comunidade(D, A, C)

1: **for** comunidade $C_i \in C$ **do**
2: Prepare os dados de treino D_i, i.e., o subconjunto de D referente aos usuários em C_i
3: Treine o modelo de recomendação M_i em D_i utilizando o algoritmo A
4: Guarde o modelo treinado M_i específico para C_i
5: **end for**
return Modelos de recomendação treinados $M_1, M_2, \cdots, M_{|C|}$ para cada comunidade

4, 6 e 9, enquanto a outra instância incluiria os usuários da segunda comunidade, a saber: usuários 2, 3, 5, 7, 8 e 10. Ao incorporar informações da comunidade no processo de treinamento, podemos aprimorar a precisão e a eficácia do modelo de recomendação proposto, visto que o modelo aprende não apenas a partir das interações individuais entre usuário e item, mas também do comportamento coletivo dentro de cada comunidade. Isso auxilia na captura de preferências sutis e na melhoria da qualidade geral das recomendações.

IV.4 Predição

Esta seção apresenta a fase de predição da metodologia de recomendação proposta. Uma vez obtida a lista de comunidades e um modelo treinado para cada comunidade, a próxima fase é gerar recomendações para qualquer usuário com base na comunidade à qual ele pertence.

Considerando um usuário u_k para o qual desejamos fornecer recomendações, podemos encontrar a comunidade C_i à qual o usuário u_k pertence. Uma vez identificada a comunidade C_i, recuperamos o modelo de recomendação treinado M_i específico para a comunidade C_i. Usando o modelo M_i, podemos gerar recomendações personalizadas para o usuário u_k. O modelo utiliza as interações usuário-item e os padrões aprendidos dentro da comunidade para fornecer recomendações adaptadas às preferências e comportamento dos usuários dessa comunidade.

O processo de previsão envolve o uso do modelo treinado M_i para calcular as classificações ou pontuações previstas para os itens com os quais o usuário u_k ainda não interagiu. Isso pode ser feito usando várias técnicas, como filtragem colaborativa, fatoração de matriz ou abordagens baseadas em aprendizado profundo. O método específico depende do algoritmo de recomendação escolhido durante a fase de treinamento. Quando as classificações ou pontuações previstas são obtidas, é possível classificar os itens ou selecionar as recomendações com melhor classificação para apresentar ao usuário u_k. Essas recomendações são baseadas nas preferências e interesses dos usuários pertencentes à mesma comunidade que u_k, capturando o comportamento coletivo e as preferências dessa comunidade.

O Algoritmo 2 resume o processo de previsão para um único usuário, recebendo como entrada um determinado usuário u_k, o conjunto de comunidades C onde $C = \{C_1, C_2, \cdots, C_{|C|}\}$, e os modelos de recomendação treinados $M = \{M_1, M_2, \cdots, M_{|C|}\}$.

Algorithm 2 : Predições Baseadas em Comunidades(u_k, C, M, k)

for cada item i não avaliado pelo usuário u_k **do**
    $score_i \leftarrow$ calcularAvaliacao(M_i, u_k, i)
    Atribuir *score* para i
end for
Ordene os items pelo *score* de maneira decrescente
Selecione os k-melhores itens como recomendações para o usuário u_k
    return k-melhores recomendações para o usuário u_k

Através da estrutura de comunidades e dos modelos de recomendação treinados, é possível fornecer recomendações personalizadas que são influenciadas não apenas pelo comportamento individual do usuário, mas também pelo comportamento dos usuários dentro de sua comunidade. Isso permite capturar as nuances e especificidades de cada comunidade, resultando em recomendações mais precisas e relevantes.

Capítulo V Avaliação Experimental

Esta seção apresenta e discute os experimentos realizados para avaliar o desempenho da metodologia de recomendação baseada em comunidades proposta. Os experimentos foram realizados em conjuntos de dados sintéticos e do mundo real para avaliar a eficácia e robustez da abordagem. Além disso, os resultados obtidos foram comparados com a abordagem padrão de alguns algoritmos de recomendação bem conhecidos para avaliar o impacto da incorporação da detecção de comunidades. Para fins de reprodutibilidade e transparência, todo o código fonte está disponível em repositório público[1].

V.1 Base de Dados

Os experimentos foram conduzidos utilizando-se tanto dados sintéticos gerados especificamente para este estudo, quanto dois conjuntos de dados do mundo real bem conhecidos, o Movielens 100k e o Movielens 1M [Harper and Konstan, 2015][2]. A escolha desses *datasets* foi motivada pela necessidade de avaliar o desempenho e a robustez da nossa metodologia de recomendação baseada em comunidades em diferentes cenários de dados.

V.1.1 Bases de Dados Movielens

Os conjuntos de dados Movielens 100k e Movielens 1M são amplamente utilizados como referências no campo de sistemas de recomendação. A criação, manutenção e atualização dessas bases de dados são de responsabilidade do GroupLens[3], o laboratório de pesquisa em sistemas de recomendações do Departamento de Ciências da Computação e Engenharia da Universidade de Minnesota[4]. Esses *datasets* contêm 10^5 e 10^6 avaliações de filmes, respectivamente. Em ambos os casos a escala de avaliação varia de 1 a 5 "estrelas", e há informações sobre os usuários e os filmes.

Para cada um desses conjuntos de dados existem três arquivos principais: *(i)* um arquivo de avaliações, *(ii)* um arquivo de informações dos usuários, e *(iii)* um arquivo de informações dos filmes. O arquivo de informações dos usuários contém detalhes sobre os usuários, como identificador do usuário, idade, sexo, ocupação e código postal. O arquivo dos filmes, por sua vez, contém

[1] Disponível em `https://github.com/moretoknow/recsys4communities`.
[2] Os dois conjuntos de dados estão disponíveis em `https://movielens.org/`.
[3] `https://grouplens.org/`
[4] `https://cse.umn.edu/cs`

informações sobre as obras cinematográficas, como identificador do filme, título, gênero e ano de lançamento. Por fim, o arquivo de avaliações – único utilizado neste trabalho – contém informações sobre as avaliações dos filmes, incluindo o identificador do usuário, o identificador do filme, a avaliação dada pelo usuário e o *timestamp* da avaliação[5]. A Tabela V.1 ilustra a estrutura dos arquivos de avaliações através de uma amostra de dez classificações referentes ao Movielens 100k.

Tabela V.1: Amostra de Avaliações do Movielens 100k

id_amostra	id_usuário	id_item	avaliação	*timestamp*
1	196	242	3	881250949
2	186	302	3	891717742
3	22	377	1	878887116
4	244	51	2	880606923
5	166	346	1	886397596
6	298	474	4	884182806
7	115	265	2	881171488
8	253	465	5	891628467
9	305	451	3	886324817
10	6	86	3	883603013

Embora os *datasets* possuam tamanhos diferentes[6], as distribuições das notas atribuídas aos filmes são bastante semelhantes, conforme ilustram as Figuras V.1 e V.2. Nelas é possível observar que parte considerável das avaliação fica entre 3 e 5, revelando uma assimetria à esquerda para ambos os casos, comportamento relativamente comum em bases de dados deste tipo.

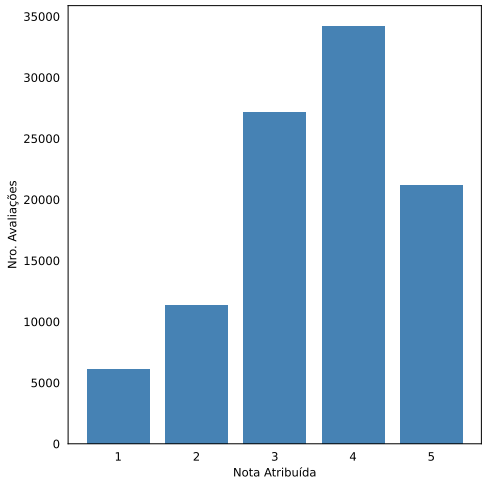


Figura V.1: Distribuição das Avaliações no Movielens-100k.

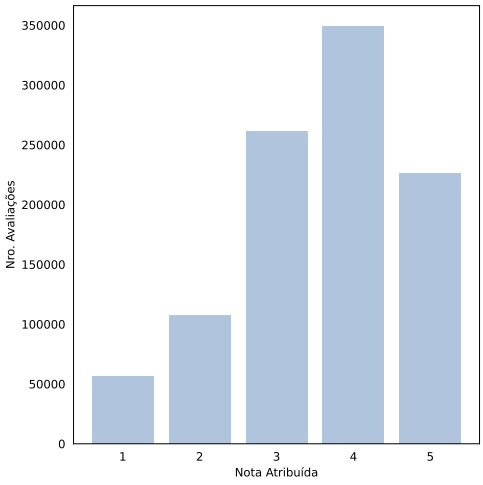


Figura V.2: Distribuição das Avaliações no Movielens-1Mk

Ademais, a esparsidade nesses conjuntos de dados é próxima de 94%, o que significa que cerca de

[5]Essas avaliações são representadas por tuplas (usuário, item, classificação, *timestamp*), e o usuário expressa suas preferências avaliando os filmes entre 1 ("não gosta") e 5 ("gosta muito").

[6]Além do maior número de avaliações, o Movielens-1M também possui seis vezes mais usuários que o Movielens-100k, além de aproximadamente três vezes mais filmes.

94% das células das matrizes de avaliações estão vazias, ou seja, não possuem valores de avaliação registrados. Isso indica que a maioria dos usuários não avaliou a maioria dos itens disponíveis no *dataset*, resultando em uma quantidade significativa de dados ausentes.

V.1.2 Bases de Dados Sintéticas

Conjuntos de dados sintéticos foram gerados para simular grafos com estruturas de comunidades. Isso permitiu estabelecer *a priori* a verdade fundamental em relação às associações de comunidades subjacentes e preferências de usuários e itens. Ao criar essas estruturas de dados sintéticas, pudemos estudar sistematicamente o impacto da detecção de comunidades no desempenho da recomendação. Esses *datasets* foram criados utilizando-se uma função personalizada que simula um grafo bipartido. A função incorpora parâmetros, como o número de usuários, comunidades, itens e categorias, além de probabilidades e intensidades de conexões entre usuários e itens. Controlando esses parâmetros, foram criados 729 conjuntos de dados sintéticos com estruturas de comunidades e padrões variados, ou seja, redes que variam no número de usuários, itens, que são mais ou menos densas e que são mais ou menos polarizadas. Para gerar cada uma das redes de recomendação comunitária, seguimos algumas etapas específicas. Essas etapas são ilustrados na Figura V.3.

Primeiro, atribuímos aleatoriamente rótulos de comunidade a cada usuário, garantindo que os rótulos estejam ordenados para manter a consistência. Isso fornece uma representação da filiação comunitária para cada usuário. Em seguida, calculamos a distribuição do tamanho da comunidade contando o número de usuários pertencentes a cada comunidade. Da mesma forma, atribuímos aleatoriamente rótulos de categoria a cada item, com rótulos ordenados para consistência. A distribuição do tamanho da categoria é então calculada com base nos rótulos de categoria atribuídos, representando a distribuição de itens em diferentes categorias.

Depois disso, é construída uma matriz de afinidade, denotada por L, para capturar a probabilidade de os usuários gostarem de itens em categorias específicas. As entradas na matriz de afinidade são geradas aleatoriamente com base em uma probabilidade α fornecida. Essa matriz quantifica o grau de afinidade entre comunidades e categorias, servindo como base para modelar as interações entre usuários e itens. Um valor mais alto de α, ou "probabilidade de gostar", implica em um maior número de conexões entre usuários de uma comunidade e itens de determinada categoria, resultando em uma rede mais densa (mais arestas), o que torna a matriz L menos esparsa – i.e., com mais avaliações disponíveis. Valores baixos de α, por outro lado, implicam em uma rede menos densa e mais esparsa.

Para levar em conta diferentes níveis de intensidade de gostar, transformamos a matriz de afinidade L em uma matriz de afinidade ajustada, denotada por L'. Os valores em L' são dimensionados pelo parâmetro de intensidade de gostar β, que controla o impacto geral da afinidade no processo

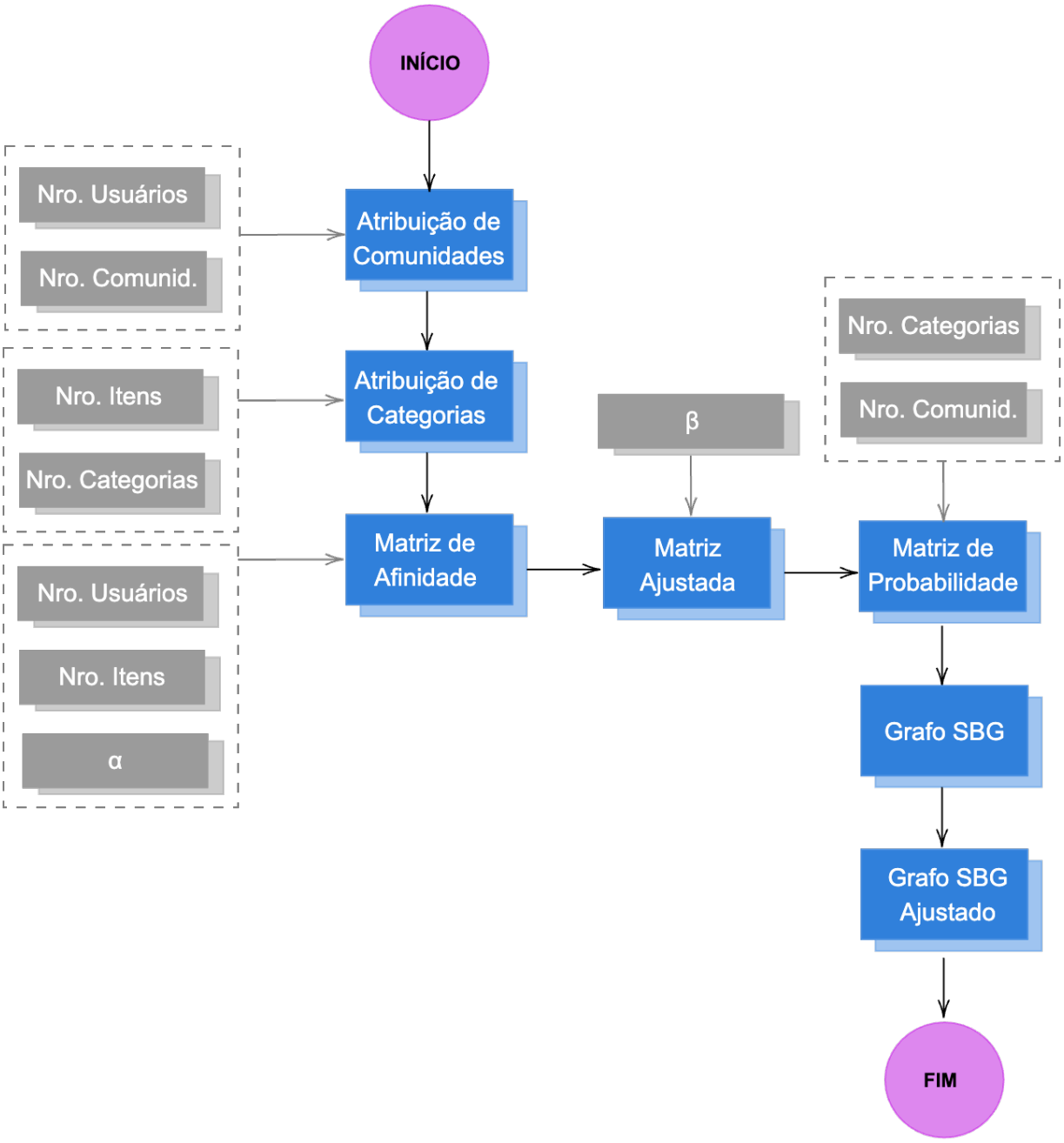


Figura V.3: Fluxograma das etapas seguidas para criação dos dados sintéticos.

de recomendação. Além disso, os valores restantes em L' são ajustados para $\frac{(1-\beta)}{2}$, garantindo uma influência equilibrada da afinidade entre comunidades e categorias. Altos valores de β, ou "intensidade de gostar", fazem com que as conexões entre os usuários e os itens que eles gostam sejam mais fortes, o que resulta em maiores pesos das arestas da rede gerada, e uma maior polarização ou especialização das preferências. Valores mais baixos de β, por sua vez, resultam em uma rede menos polarizada.

Ao combinar a matriz de afinidade ajustada L' com matrizes de zeros, construímos uma matriz de probabilidade P. Essa matriz captura as probabilidades de interações entre usuário e item com

base nas filiações de comunidade e categoria. Utilizando as distribuições de usuário e item, bem como a matriz de probabilidade P, criamos um *Stochastic block model (SBG)* usando a biblioteca NetworkX [Hagberg et al., 2008]. O SBG representa o grafo de recomendação, onde os tamanhos dos blocos correspondem às distribuições de usuários e itens. O grafo de modelo de blocos estocásticos é uma representação do grafo de recomendação, onde os nós representam os usuários e itens, e as arestas representam as interações entre eles.

Após a geração do grafo de recomendação, as avaliações de itens são atribuídas aos usuários com base nas probabilidades de interação calculadas anteriormente. Essas avaliações representam as preferências dos usuários pelos itens no contexto das suas comunidades e categorias correspondentes. Por fim, as avaliações são normalizadas como um inteiro dentro do intervalo $[1, 5]$ para emular as faixas de avaliações comumente observadas em *datasets* do mundo real.

Com o grafo de recomendação e as avaliações de itens, podemos realizar experimentos para avaliar o desempenho da metodologia de recomendação baseada em comunidades. Isso envolve a execução de algoritmos de detecção de comunidades no grafo para identificar as comunidades subjacentes. Em seguida, os modelos de recomendação são treinados com base nas avaliações de itens de cada comunidade. Posteriormente, podemos realizar previsões de recomendação para usuários específicos com base nas comunidades a que pertencem, utilizando os modelos de recomendação treinados.

Para fins experimentais, definimos três opções de valores para cada um dos seis parâmetros da função geradora de dados sintéticos, sendo eles:

- **Número de usuários**: 50, 100 e 200;
- **Número de comunidades**: 5, 10 e 20;
- **Número de itens**: 50, 100 e 200;
- **Número de categorias**: 5, 10 e 20;
- **Probabilidade de Gostar (α)**: 20%, 40%, e 60%;
- **Intensidade (β)**: 20%, 50%, e 80%.

A combinação desses valores resultou em $3^6 = 729$ redes sintéticas de diferentes características. Essas opções foram baseadas na estrutura padrão da função criada. Isto é, se nenhuma informação for passada no momento da criação da rede, será considerada uma estrutura com 100 usuários, 100 itens, 10 categorias e 10 itens. Além disso, as probabilidades α e β foram definidas em 0.4 e 0.5, respectivamente. Dessa maneira, para fins de variabilidade, definiram-se opções de valores acima e abaixo desses números *default* para cada parâmetro.

Através desses experimentos, podemos avaliar a precisão e eficácia da metodologia de recomendação baseada em comunidades em comparação com outras abordagens. Também podemos analisar a influência dos parâmetros de afinidade e intensidade de gostar na qualidade das recomendações, além das questões mais associadas à estrutura da rede, como número de usuários, comunidades, itens e categorias. Esses experimentos nos ajudam a entender o impacto da detecção de comunidades no desempenho dos sistemas de recomendação e a explorar as vantagens de capturar a estrutura de comunidades na geração de recomendações personalizadas.

V.2 Protocolo Experimental

Para avaliar o desempenho da nossa metodologia de recomendação baseada em comunidades, dividimos cada um dos 731 conjuntos de dados[7] em conjuntos de treinamento e teste. Foram considerados múltiplos tamanhos de teste, incluindo 25%, 10% e 1% do total de avaliações. Para garantir avaliações confiáveis e representativas, foi empregada uma técnica de validação cruzada com permutação aleatória. Essa técnica envolve permutar aleatoriamente o conjunto de dados e, em seguida, dividir em vários subconjuntos de treinamento e teste. Ao usar essa abordagem, é possível mitigar possíveis vieses que possam surgir da ordem ou distribuição dos dados. Nos experimentos, foi utilizada a implementação de validação cruzada com permutação aleatória fornecida pela biblioteca scikit-learn [Pedregosa et al., 2011]. O número de divisões foi definido como cinco, permitindo a geração de vários subconjuntos de treinamento e teste. Ao empregar essa técnica, buscamos obter uma avaliação robusta do desempenho dos nossos modelos de recomendação em diferentes partições de dados.

Para cada instância de treinamento, o processo de recomendação consiste em várias etapas. Primeiro, o grafo bipartido é transformado em um grafo projetado, conforme descrito na Seção IV.1. Esse grafo projetado captura as interações usuário-item e permite o foco nas relações usuário-usuário e item-item. Em seguida, os algoritmos de detecção de comunidades são aplicados para identificar comunidades dentro do grafo projetado. Nos experimentos realizados, foram utilizados os algoritmos Louvain [Blondel et al., 2008c] e Paris [Bonald et al., 2018; Xie et al., 2013] para detecção de comunidades, pois são amplamente utilizados na literatura. Esses algoritmos particionam o grafo em comunidades distintas com base nos padrões de conectividade entre usuários e itens.

Após a identificação das comunidades, os modelos de recomendação são instanciados e treinados especificamente para cada comunidade. Foram avaliados cinco algoritmos de recomendação diferentes, disponíveis na biblioteca Surprise [Hug, 2020]. Esses algoritmos representam um conjunto diversificado de técnicas comumente usadas em sistemas de recomendação, agrupados em três categorias:

[7] Número total de *datasets*, incluindo 729 dados sintéticos, Movielens-100k e Movielens-1M.

1. **Algoritmos de fatoração matricial**
   (a) *Singular Value Decomposition* (*SVD*)
   (b) *Non-Negative Matrix Factorization (NMF)*
2. **Algoritmos baseados em vizinhos**
   (a) *k-Nearest Neighbors* (*k-NN*)
   (b) *Slope One*
3. **Algoritmos de clusterização**
   (a) *Co-Clustering*

Além disso, o algoritmo *Normal Predictor* foi incluído como um *baseline* aleatório. Para dado item i ainda não avaliado pelo usuário u_k, esse algoritmo prevê uma avaliação aleatória $\hat{r}_{u_k,i}$ com base na distribuição do conjunto de treinamento $\mathcal{N}(\hat{\mu}, \sigma^2)$, onde $\hat{\mu}$ e σ^2 são estimados a partir dos dados de treinamento usando a Estimação de Máxima Verossimilhança.

Uma vez que os modelos foram treinados, o processo de previsão foi aplicado a cada par usuário-item nos dados de teste. Seguindo a abordagem descrita na Seção IV.4, os modelos treinados com base nas estrutura comunitária foram utilizandos nas comunidades correspondentes para gerar previsões para as avaliações.

Para avaliar o desempenho da abordagem proposta nesta pesquisa, utilizamos a métrica de RMSE, que fornece uma medida quantitativa da eficácia dos modelos de recomendação na previsão das avaliações usuário-item. A escolha da medida se deu tanto pela facilidade de cálculo através da biblioteca *Surprise*, quanto pela vasta utilização em trabalhos anteriores. Além disso, o custo computacional da abordagem foi avaliado, incluindo os tempos de treinamento e teste em segundos. Essa análise permitiu avaliar não apenas a precisão, mas também a eficiência da abordagem de recomendação baseada em comunidades proposta.

Para além do fluxo experimental, a infraestrutura computacional utilizada para a condução dos experimentos foi um sistema Linux com um processador Intel Xeon Gold 5215 CPU @ 2.50GHz, dotado de 16 núcleos, operando em modos tanto de 32 bits quanto de 64 bits e 64 GB de memória RAM.

Ao realizar experimentos em conjuntos de dados sintéticos e do mundo real, nosso objetivo foi fornecer uma avaliação abrangente da nossa abordagem de recomendação baseada em comunidades e avaliar seu desempenho em diferentes cenários de dados. Os resultados obtidos desses experimentos são apresentados e discutidos na próxima subseção.

V.3 Resultados e Discussões

Esta seção apresenta os resultados dos experimentos computacionais realizados para avaliar o desempenho da abordagem de recomendação baseada em comunidades. A abordagem proposta é comparada com os métodos de recomendação padrão, isto é, a nível global e não a nível comunitário. A eficácia e o custo computacional em diferentes cenários são analisados.

Para avaliar o desempenho da abordagem de recomendação baseada em comunidades, os testes foram realizados utilizando conjuntos de dados sintéticos e reais, como elucidado anteriormente (Seção V.1).

V.3.1 Desempenho

A Figuras V.4 ilustra a comparação de desempenho entre a abordagem proposta, utilizando os algoritmos Louvain e Paris no processo de descoberta de comunidades, e a alternativa padrão, que corresponde ao processo sem a utilização de algoritmo de descoberta de comunidades, em dados reais e na versão *default*[8] dos dados sintéticos considerando seis diferentes algoritmos de recomendação, a saber, *i) Co-Clustering*, *ii)* NMF, *iii) Normal Predictor*, *iv)* SVD, *v) Slope One* e *vi)* k-NN. As alturas das barras correspondem ao RMSE médio das iterações realizadas, enquanto que as faixas de erro representam a variabilidade ou incerteza associada a esses valores médios. Essa variabilidade foi calculada utilizando o desvio padrão dos resultados obtidos em cada iteração.

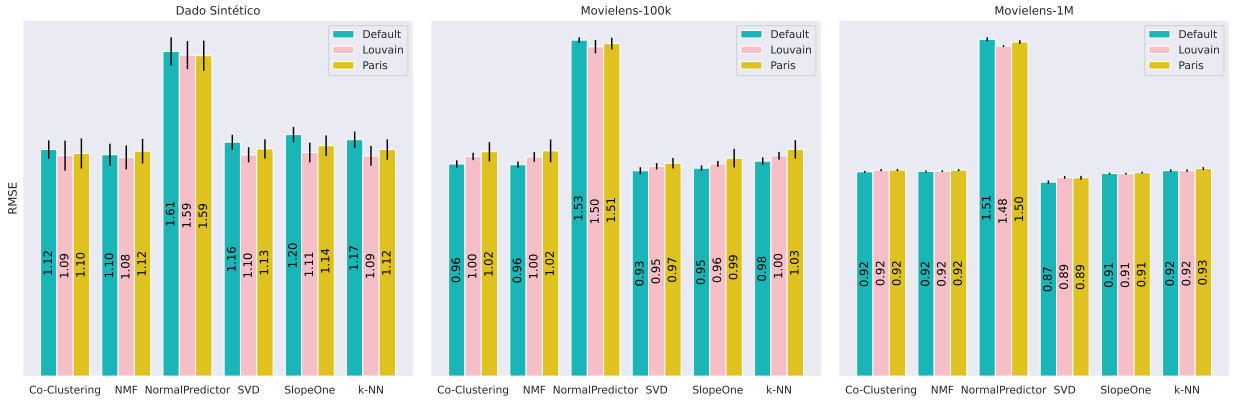


Figura V.4: Desempenho em RMSE considerando diferentes conjuntos de dados, abordagens de detecção de comunidades e algoritmos de recomendação.

Considerando o conjunto de dados sintéticos com configurações *default*, a abordagem de recomendação baseada em comunidades, proposta, demonstrou resultados promissores para todos os tamanhos de amostras de teste. Ela consistentemente alcançou valores de RMSE mais baixos em comparação com o método padrão (sem detector de comunidades) para quase todos os algoritmos.

[8]Devido ao grande número de registros, optou-se por exibir apenas a versão que que faz uso dos parâmetros *default* da função geradora para construir uma rede sintética, a saber: 100 usuários, 100 itens, 10 comunidades, 10 categorias, $\alpha = 0.4$ e $\beta = 0.5$ (Subseção V.1.2). No entanto, resultados para todas as versões estão dispostas no repositório.

Uma ênfase especial pode ser dada aos algoritmos baseados em vizinhança, que apresentaram a maior melhoria de desempenho quando combinados com a detecção de comunidades.

Ao examinar o cenário de conjunto de dados reais, nossa abordagem mostrou desempenho semelhante ao método padrão para quase todos os algoritmos, embora ligeiramente pior. A única exceção é o algoritmo *Normal Predictor*, que apresentou um RMSE menor em cenários em que os detectores de comunidade são usados. Entretanto, mesmo para conjuntos de dados reais, o desempenho geral da abordagem proposta parece melhorar com o aumento do tamanho dos dados, uma vez que os resultados obtidos no Movielens-1M são mais promissores que aqueles obtidos no Movielens-100k.

A Figura V.5 ilustra esse processo de tendência de melhoria do desempenho da abordagem proposta ao utilizar-se um conjunto de dados maior. Nela é possível verificar a DP entre os RMSEs obtidos com e sem a utilização de estruturas comunitárias ($\mathrm{DP}_{\mathrm{RMSE}}$). Essa métrica é calculada à partir da Equação (V.1), onde $\mathrm{RMSE}_{\mathrm{controle}}$ é o RMSE médio obtido sem a utilização de detectores de comunidades, e $\mathrm{RMSE}_{\mathrm{tratamento}}$ é o argumento máximo entre os RMSEs obtidos com o a utilização de Louvain e Paris. Valores negativos de $\mathrm{DP}_{\mathrm{RMSE}}$ indicam queda relativa do RMSE ao utilizar-se detectores de comunidades – *coeteris paribus*. Valores positivos dessa métrica, por sua vez, indicam piora do desempenho ao incluir detectores de comunidade no *pipeline*.

$$\mathrm{DP}_{\mathrm{RMSE}} = \left(\frac{\mathrm{RMSE}_{\mathrm{tratamento}} - \mathrm{RMSE}_{\mathrm{controle}}}{\mathrm{RMSE}_{\mathrm{controle}}} \right) \times 100 \tag{V.1}$$

Assim, é possível notar que embora os algoritmos de *Co-Clustering*, NMF, *NormalPredictor*, *Slope One* e k-NN tenham apresentado crescimento do RMSE ao utilizar a abordagem proposta em todos os conjuntos de dados reais, o aumento decresce consideravelmente quando considerado o Movielens-1M, o que indica a existência teórica de um ponto de inflexão à partir do qual teríamos DPs negativas com o aumento dos conjuntos de dados. Esse fato sugere maior aderência da abordagem proposta em cenários onde há um maior número de usuários e avaliações disponíveis.

Cenário semelhante pode ser observado quando considerados apenas os dados sintéticos. A Figura V.6 mostra como os 25 piores e os 25 melhores resultados em termos de $\mathrm{DP}_{\mathrm{RMSE}}$ se distribuíram quando considerados diferentes algoritmos de recomendação (eixo vertical da figura) e parâmetros das redes sintéticas (eixo horizontal da figura). Para cada um dos seis modelo sob análise, foram calculados a $\mathrm{DP}_{\mathrm{RMSE}}$ de cada iteração (*dataset*), e em seguida, as iterações foram ordenadas com base nessa métrica. Após essa ordenação, foram selecionados os 25 melhores e os 25 piores desempenhos. No caso dos 25 melhores, trata-se dos *datasets* que obtiveram a menor $\mathrm{DP}_{\mathrm{RMSE}}$, indicando maior ganho relativo ao utilizar a abordagem proposta. O raciocínio inverso se aplica aos 25 piores registros. Esse procedimento permitiu uma análise detalhada para compreender padrões, comportamentos ou características associadas a esses extremos

Pode-se observar que para parte considerável dos algoritmos as distribuições dos melhores resul-

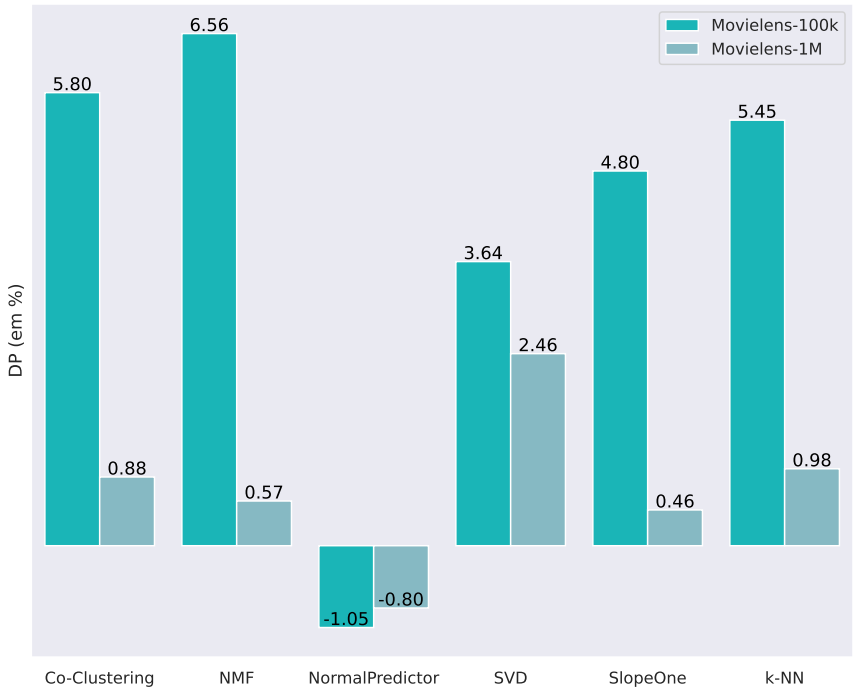


Figura V.5: DP entre o RMSE da abordagem proposta e o da abordagem padrão.

tados concentraram-se em cenários com alto número de usuários e itens, enquanto que os conjuntos dos piores resultados concentraram-se em baixos valores desses parâmetros. Esse cenário fica mais evidente quando considerados os algoritmos baseados em vizinhos ou em clusterização, onde houve maior assimetria ou contraste entre as distribuições dos 25 melhores e 25 piores cenários. Além disso, foi também possível observar alta divergência das distribuições no que diz respeito ao número de comunidades e à intensidade de gostar β, onde para todos os algoritmos houve maior concentação dos melhores casos em cenários de menor número de estruturas comunitárias e menor polarização, e maior convergência dos piores casos no cenário oposto (maior número de comunidades polarização). Algoritmos baseados em vizinhos e em cluterização foram aquele que obtiveram maior discrepância das distribuições quando considerados esses parâmetros. Por fim, não foram observadas contraposições relevantes entre as distribuições dos melhores e piores casos quando considerados o número de categorias e a probabilidade de gostar α para a maior parte dos algoritmos. A única excessão é o algoritmos *Normal Predictor*, que apresentou maior concentração dos melhores casos em cenários com poucas categorias distintas.

Essas descobertas sugerem que a utilização de detectores de comunidade parece ser benéfica para tarefas de recomendação, especialmente para conjuntos de dados com estruturas de rede mais bem definidas, como é o caso da rede sintética *default*, onde foram observados maiores ganhos em termos

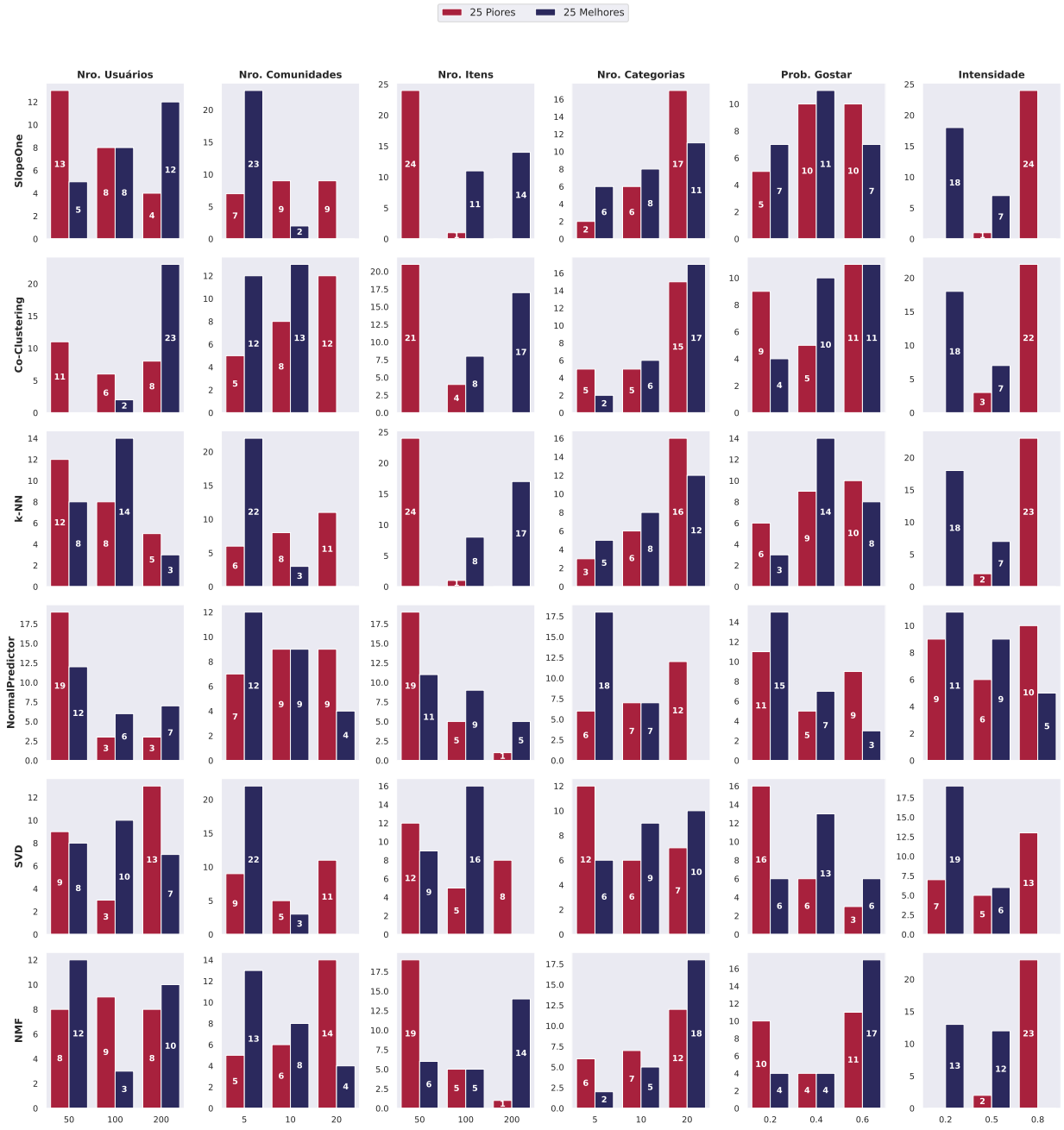


Figura V.6: Distribuição dos 25 Melhores e 25 Piores casos em termos da DP de RMSE.

de RMSE. No entanto, mesmo para casos com estruturas pouco modulares – como os conjuntos de dados reais analisados – a abordagem pode trazer ganhos em termos de RMSE se houver uma alta disponibilidade de dados – i.e., muitas avaliações, usuários e itens. Ademais, redes com um menor número de estruturas comunitárias e menor grau de polarização podem se beneficiar de uma maneira ainda maior do método proposto. Por fim, os resultados indicam que os algoritmos baseados em vizinhança demonstraram maior sinergia com nossa abordagem de recomendação baseada em comunidades.

V.3.2 Custo Computacional

Além da avaliação de desempenho, o custo computacional da abordagem de recomendação baseada em comunidades também foi analisado. Dois aspectos foram considerados: *i)* tempo de treinamento e *ii)* tempo de predição. A Figura V.7 mostra o tempo de treinamento para cada algoritmo em diferentes cenários, tanto para o conjunto de dados sintéticos *default* quanto dados reais. As alturas das barras correspondem à mediana dos tempos de execuções verificados em cada iteração, enquanto que as faixas de erro representam a variabilidade ou incerteza associada a esses valores medianos. Essa variabilidade foi calculada utilizando o *Median Absolute Deviation (MAD)*. O tempo mostrado não inclui aquele despendido no processo de detecção das comunidades, que encontra-se discriminado na Tabela V.2

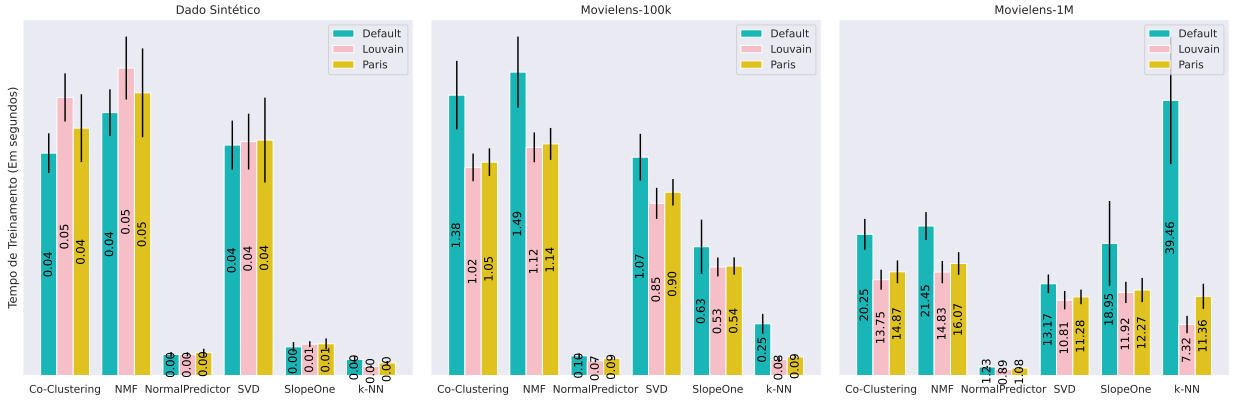


Figura V.7: Tempo de treinamento considerando diferentes conjuntos de dados, abordagens de detecção de comunidades e algoritmos de recomendação.

Tabela V.2: Tempo despendido no processo de detecção de comunidades.

Conjunto de Dados	Detector	Tempo (em seg.)
Movielens-100k	Louvain	8.345428
	Paris	0.154182
Movielens-1m	Louvain	443.224014
	Paris	10.517705
Sintético	Louvain	0.057697
	Paris	0.000381

Embora os resultados para o dado sintético *default* tenham mostrado um aumento marginal no tempo despendido em treinamento, a utilização de detectores reduziu significativamente esse custo para todos os algoritmos nos cenários de dados reais. Além disso, foi possível obeservar uma melhora do desempenho à medida que se utilizaram dados maiores. A Figura V.8 ilustra essa melhoria em termos da DP do tempo de execução (DP_{TEMPO}). Essa métrica é definida pela Equação (V.2), onde $TEMPO_{controle}$ é a mediana do tempo de treinamento obtido sem a utilização de detectores de comunidades, e $TEMPO_{tratamento}$ é o argumento máximo entre as medianas dos tempos de

treinamento obtidos com o a utilização de Louvain e Paris[9]. Valores negativos de $\mathrm{DP}_{\mathrm{TEMPO}}$ indicam queda relativa do tempo de treinamento ao utilizar-se detectores de comunidades, enquanto que valores positivos dessa métrica indicam aumento do custo computacional ao incluir detectores de comunidade.

$$\mathrm{DP}_{\mathrm{TEMPO}} = \left(\frac{\mathrm{TEMPO}_{\mathrm{tratamento}} - \mathrm{TEMPO}_{\mathrm{controle}}}{\mathrm{TEMPO}_{\mathrm{controle}}} \right) \times 100 \tag{V.2}$$



Figura V.8: DP entre o tempo de treino da abordagem proposta e o da abordagem padrão.

No que se refere aos dados sintéticos, embora na versão padrão deste conjunto de dados não tenham sido observados ganhos de $\mathrm{DP}_{\mathrm{TEMPO}}$, a Figura V.9 mostra que todos os algoritmos analisados apresentaram uma alta disparidade das distribuições dos melhores e piores DPs de tempo de treinamento, com uma grande concentração dos melhores resultados em cenários com um maior número de usuários e itens. Além disso, foi também possível observar alta divergência das distribuições no que diz respeito ao número de comunidades para os algoritmos *Normal Predictor*, NMF, *Slope One* e *Co-Clustering*, onde houve maior concentação dos melhores casos em cenários de menor número

[9]Não inclui o tempo gasto no processo de detecção das comunidades.

de estruturas comunitárias, e maior convergência dos piores casos na conjuntura oposta. Adicionalmente, contraposições relevantes entre as distribuições das melhores e piores iterações quando considerada a probabilidade de gostar α foram identificadas para os algoritmos baseados em vizinhos, com maior concentração dos melhores casos em circunstâncias de redes mais densas. Por fim, o algoritmo k-NN demonstrou melhor desempenho geral em contextos de menor polarização, enquanto as abordagens NMF e *Co-Clustering* tiveram maior concentração de iterações ruins em contextos envolvendo um maior número de categorias de itens.

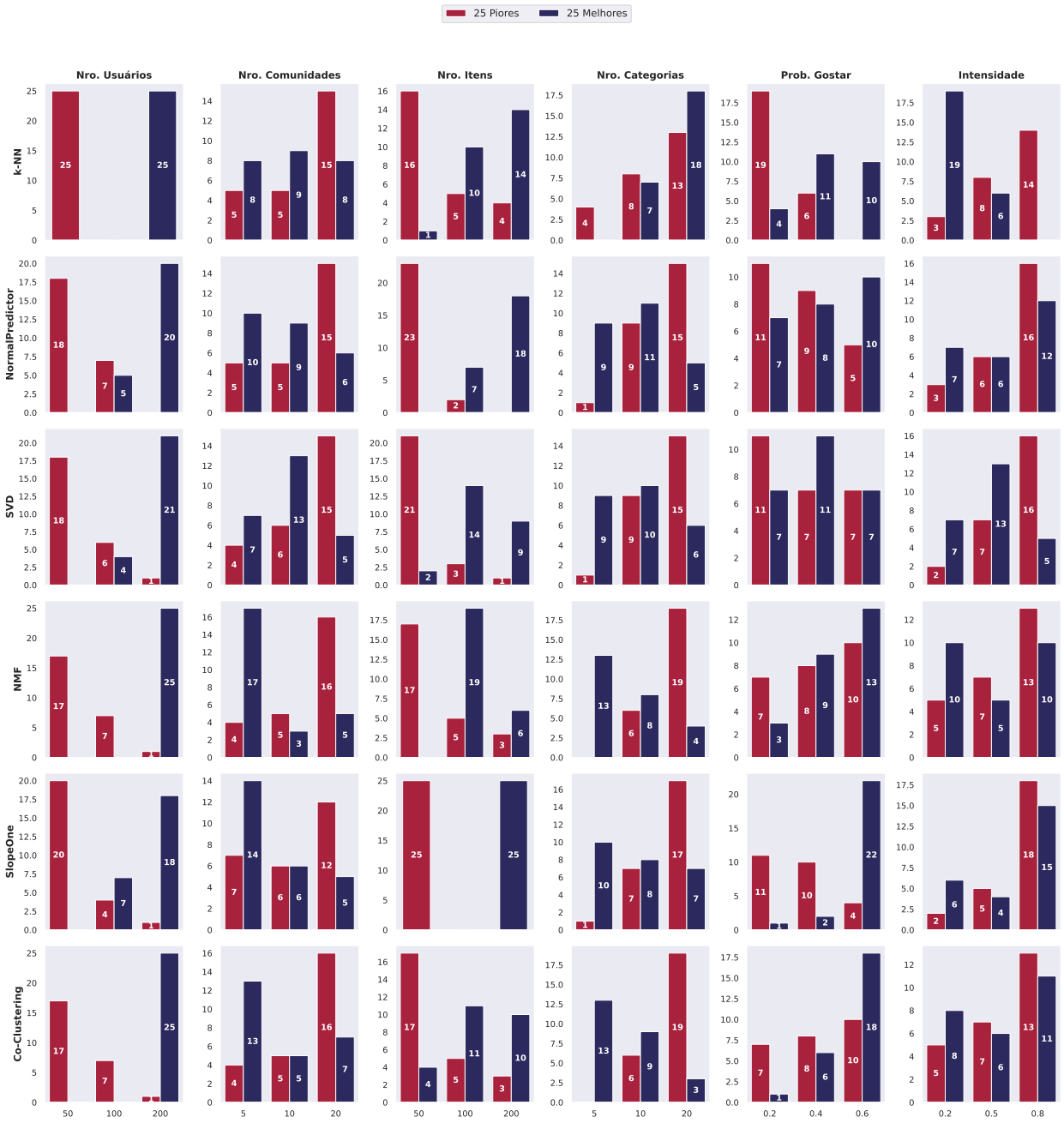


Figura V.9: Distribuição dos 25 Melhores e 25 Piores casos em termos da DP do tempo de treino.

Importante ressaltar que embora o tempo despendido em treinamento tenha diminuído com o aumento dos dados, processo oposto ocorreu com a etapa de detecção das comunidades, onde houve aumento do tempo em cenários de maiores informações – mais usuários, itens ou avaliações. Nesse sentido, em alguns casos o tempo gasto na etapa de detecção tornou o tempo total[10] maior do que aquele obtido nas execuções sem detecção de comunidades, sobretudo ao utilizar-se o algoritmo de Louvain, que possui maior complexidade que a abordagem de Paris. Entretanto, mesmo utilizando-se Louvain em grandes estruturas de dados, foi possível ainda assim obter ganhos frente às estratégias padrão mesmo ao somar-se o tempo gasto com a detecção counitária, sobretudo utilizando algoritmos baseados em vizinhos.

A Figura V.10, por sua vez, apresenta a mesma visão aplicada ao tempo de predição. As alturas das barras correspondem à mediana dos tempos de execuções verificados em cada iteração, enquanto que as faixas de erro representam a variabilidade ou incerteza associada a esses valores medianos. Essa variabilidade foi calculada utilizando o MAD.

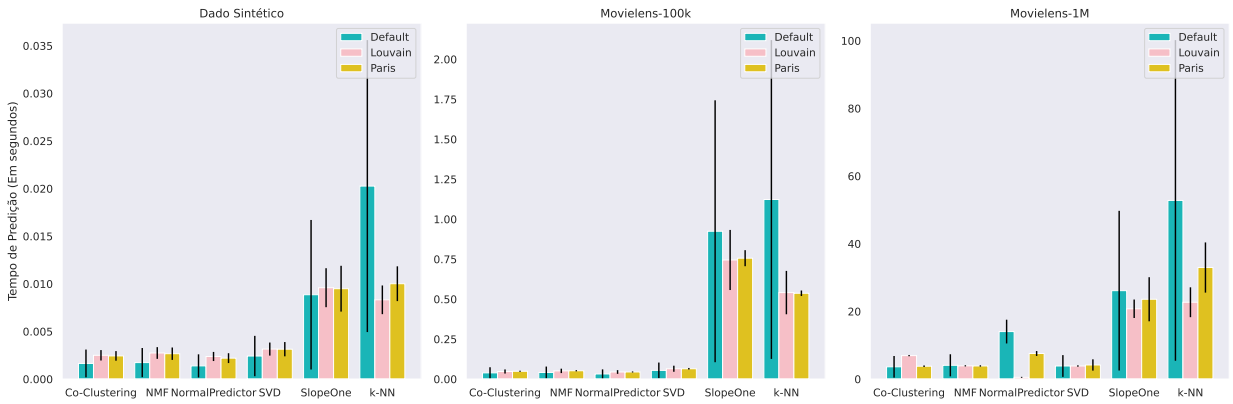


Figura V.10: Tempo de predição considerando diferentes conjuntos de dados, abordagens de detecção de comunidades e algoritmos de recomendação.

A utilização de detectores de comunidade diminuiu significativamente o tempo despendido em predições para algoritmos baseados em vizinhos. Para demais métodos, a abordagem proposta manteve competitividade.

Em resumo, a abordagem proposta neste trabalho se apresenta promissora em melhorar o desempenho dos algoritmos de recomendação, especialmente para grandes conjuntos de dados com muitos usuários e itens disponíveis. Além disso, a existência de estruturas comunitárias implícitas e pouco polarizadas pode aprimorar ainda mais o desempenho em termos de RMSE. Ademais, a abordagem reduziu significativamente os custos computacionais, sobretudo quando utilizou algoritmos menos complexos para execução das detecções, como Paris. Por fim, algoritmos baseados em vizinhos demonstraram maior sinergia com a abordagem, uma vez que obtiveram maior ganho de performance em termos de RMSE e custo computacional ao utilizar detectores de comunidades. Essas desco-

[10] Tempo de treinamento acrescido do tempo gasto com a etapa de detecção.

bertas destacam o potencial dessa abordagem em sistemas de recomendação, demonstrando sua promissora capacidade de melhorar o desempenho enquanto otimiza os recursos computacionais.

Capítulo VI Considerações Finais

Os algoritmos de FC são os mais utilizados na indústria para construção de sistemas de recomendação diversos. No entanto, quando aplicados a conjuntos de dados comerciais modernos, essas abordagens frequentemente enfrentam problemas de desempenho decorrentes da esparsidade. Para lidar com essas questões, abordagens variadas têm sido discutidas na literatura, como a incorporação de informações adicionais para modelar as preferências dos usuários, a inclusão de novas medidas de similaridade e, sobretudo, a adoção de abordagens que utilizem algoritmos de detecção de comunidades ou agrupamento para realizar inferências nos níveis locais dessas redes.

Este trabalho propôs conduzir uma investigação comparativa para avaliar como os algoritmos de FC podem alcançar um maior desempenho estatístico e/ou computacional quando combinados às técnicas de detecção de comunidades, com ênfase aos detectores de comunidade de Louvain e Paris. Para tanto, foi providenciada uma estrutura comum de experimentação, na qual os algoritmos de FC foram executados e tiveram seus desempenhos avaliados por meio de diversas métricas estabelecidas. Dados reais e sintéticos foram utilizados para fins experimentais.

Como resultado, observou-se que a abordagem baseada em comunidades proposta apresenta resultados promissores, especialmente em cenários de dados sintéticos com estruturas de rede bem definidas. Ela consistentemente superou o método padrão em termos de valores de RMSE mais baixos, enfatizando seu potencial para melhorar o desempenho de recomendação. Além disso, a sinergia entre os detectores de comunidade e os algoritmos baseados em vizinhança realça ainda mais as vantagens de nossa abordagem. Mesmo em conjuntos de dados sem *clusters* claros, nossa abordagem mantém um desempenho comparável ao método padrão, ao mesmo tempo que reduz significativamente os custos computacionais. Além disso, a abordagem sugere ganhos consistentes, em termos de RMSE e custo computacional, se forem considerados grandes conjuntos de dados. Essas descobertas fornecem informações valiosas para pesquisadores e profissionais na área de sistemas de recomendação, destacando o potencial das abordagens baseadas em comunidades para melhorar a precisão de recomendação e otimizar a eficiência computacional.

Além das descobertas e análises realizadas, um resultado significativo deste trabalho foi a proposição de um modelo para a geração de conjuntos de dados sintéticos. Esta abordagem mostrou-se crucial para a realização de experimentos controlados e a avaliação de algoritmos de recomendação. A função desenvolvida permitiu a criação de conjuntos de dados sintéticos que reproduzem cenários

complexos, simulando diferentes perfis de redes, contribuindo assim para a ampliar as possibilidades de experimentação. A capacidade de gerar dados sintéticos oferece um caminho promissor para futuros estudos, permitindo a análise de algoritmos de recomendação em condições controladas e a validação de modelos preditivos em situações simuladas.

Extrapolações futuras podem ser feitas a fim de se verificar se outros algoritmos ou combinações de parâmetros podem apresentar melhora semelhante ou superior quando submetidos ao mesmo *pipeline*. Nesse sentido, métricas adicionais podem ser incorporadas ao fluxo de avalição, incluindo àquelas mais relacionadas ao negócio, como o Return on Investments (ROI). Além disso, novos conjuntos de dados podem ser utilizados na intenção de expandir os experimentos até então executados, incluindo outros conjuntos de dados reais e dados sintéticos adicionais com diferentes características. Por fim, abordagens que levem em conta comunidades sobrepostas também podem ser avaliadas no futuro.

Referências

Abbasi, A. and Altmann, J. On the correlation between research performance and social network analysis measures applied to research collaboration networks. In *2011 44th Hawaii international conference on system sciences*, pages 1–10. IEEE, 2011.

Abdrabbah, S. B., Ayachi, R., and Amor, N. B. A dynamic community-based personalization for e-government services. In *Proceedings of the 9th International Conference on Theory and Practice of Electronic Governance*, pages 258–265, 2016.

Adomavicius, G. and Tuzhilin, A. Expert-driven validation of rule-based user models in personalization applications. *Data Mining and Knowledge Discovery*, 5(1):33–58, 2001.

Adomavicius, G. and Tuzhilin, A. Toward 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(6):734–749, 2005.

Airoldi, E. M., Blei, D., Fienberg, S., and Xing, E. Mixed membership stochastic blockmodels. *Advances in neural information processing systems*, 21, 2008.

Al-Bakri, N. F. and Hashim, S. H. Reducing data sparsity in recommender systems. *Al-Nahrain Journal of Science*, 21(2):138–147, 2018.

Anyosa, S. C., Vinagre, J., and Jorge, A. M. Incremental matrix co-factorization for recommender systems with implicit feedback. In *Companion Proceedings of the The Web Conference 2018*, pages 1413–1418, 2018.

Balabanović, M. and Shoham, Y. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

Barnard, S. T., Pothen, A., and Simon, H. D. A spectral algorithm for envelope reduction of sparse matrices. *Numerical Lin. Alg. with Applic.*, 2(4):317–334, 1995.

Bawden, D., Holtham, C., and Courtney, N. Perspectives on information overload. In *Aslib proceedings*. MCB UP Ltd, 1999.

Bennett, J. and Lanning, S. The netflix prize. In *Proceedings of the KDD Cup Workshop 2007*, pages 3–6, New York. ACM, 2007.

Biggs, N., Lloyd, and Wilson, R. J. *Graph theory, 1736-1936*. Clarendon Press, 1986.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008a.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks, 2008b.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008c.

Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. Recommender systems survey. *Knowledge-based systems*, 46:109–132, 2013.

Bonald, T., Charpentier, B., Galland, A., and Hollocou, A. Hierarchical graph clustering using node pair sampling. *arXiv preprint arXiv:1806.01664*, 2018.

Borgatti, S. P. and Halgin, D. S. Analyzing affiliation networks. *The Sage handbook of social network analysis*, 1:417–433, 2011.

Bourhim, S., Benhiba, L., and Idrissi, M. A. J. Towards a novel graph-based collaborative filtering approach for recommendation systems. In *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, SITA'18, New York, NY, USA. Association for Computing Machinery, 2018a.

Bourhim, S., Benhiba, L., and Idrissi, M. J. Towards a novel graph-based collaborative filtering approach for recommendation systems. In *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, pages 1–6, 2018b.

Bourhim, S., Benhiba, L., and Idrissi, M. J. Investigating algorithmic variations of an rs graph-based collaborative filtering approach. In *Proceedings of the ArabWIC 6th Annual International Conference Research Track*, pages 1–6, 2019.

Broder, A. Z. On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE, 1997.

Burke, R. Knowledge-based recommender systems. *Encyclopedia of library and information systems*, 69(Supplement 32):175–186, 2000.

Burke, R. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

Burke, R. Hybrid web recommender systems. *The Adaptive Web*, pages 377–408, 2007.

Cao, K. Y., Liu, Y., and Zhang, H. X. Improving the cold start problem in music recommender systems. In *Journal of Physics: Conference Series*, volume 1651, page 012067. IOP Publishing, 2020.

Castells, M. *The network society: A cross-cultural perspective*. Edward Elgar Publishing, Incorporated, 2004.

Cazabet, R. and Amblard, F. Simulate to detect: a multi-agent system for community detection. In *2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 2, pages 402–408. IEEE, 2011.

Chen, C.-W., Lamere, P., Schedl, M., and Zamani, H. Recsys challenge 2018: Automatic music playlist continuation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 527–528, 2018.

Chen, L., Chen, G., and Wang, F. Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction*, 25(2):99–154, 2015.

Choudhary, C., Singh, I., and Kumar, M. Community detection algorithms for recommendation systems: techniques and metrics. *Computing*, 105(2):417–453, 2023.

Çiftçi, O., Tenekeci, S., et al. Artist recommendation based on association rule mining and community detection. In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management-KDIR*. SCITEPRESS, 2021.

Clauset, A., Newman, M. E., and Moore, C. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. Comparing community structure identification. *Journal of statistical mechanics: Theory and experiment*, 2005(09):P09008, 2005.

Das, A. S., Datar, M., Garg, A., and Rajaram, S. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280, 2007.

Dias, P. From 'infoxication'to 'infosaturation': a theoretical overview of the cognitive and social effects of digital immersion. *ambitos*, 24, 2014.

Ding, C. A Tutorial on Spectral Clustering Spectral Gold-Rush of 2001 Part I : Basic Theory , 1973 – 2001. *Energy*, 2004.

Dugué, N. and Perez, A. *Directed Louvain: maximizing modularity in directed networks*. PhD thesis, Université d'Orléans, 2015.

Dunn, J. C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 1973.

Fortunato, S. and Barthelemy, M. Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41, 2007.

Fortunato, S. and Castellano, C. Community structure in graphs, 2007.

Ganji, M., Seifi, A., Alizadeh, H., Bailey, J., and Stuckey, P. J. Generalized modularity for community detection. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II 15*, pages 655–670. Springer, 2015.

Girvan, M. and Newman, M. E. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Girvan, M. and Newman, M. E. J. Community structure in social and biological networks, 2001.

Golbeck, J. Generating predictive movie recommendations from trust in social networks. In *International Conference on Trust Management*, pages 93–104. Springer, 2006.

Goodall, B. A. The managing of organizations: The administrative struggle, 1967.

Gorripati, S. K. and Vatsavayi, V. K. Community-based collaborative filtering to alleviate the cold-start and sparsity problems. *International Journal of Applied Engineering Research*, 12(15):5022–5030, 2017.

Gross, B. M. The managing of organizations: The administrative struggle. *(No Title)*, 1964.

Guimera, R. and Amaral, L. A. N. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

Gunawardana, A. and Meek, C. A unified approach to building hybrid recommender systems. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 117–124, New York, NY, USA. ACM, 2009.

Guo, G. Improving the performance of recommender systems by alleviating the data sparsity and cold start problems. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

Hadil, A. and Mandi, N.-E.-Y. *Système de recommandation à base de communautés*. PhD thesis, Université Ibn Khaldoun-Tiaret-, 2022.

Hagberg, A., Swart, P., and S Chult, D. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

He, C., Parra, D., and Verbert, K. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 56:9–27, 2016.

Himma, K. E. and Tavani, H. T. *The handbook of information and computer ethics*. Wiley Online Library, 2008.

Hug, N. Surprise: A python library for recommender systems. *Journal of Open Source Software*, 5(52):2174, 2020.

Jacoby, J. Perspectives on information overload. *Journal of consumer research*, 10(4):432–435, 1984.

Jain, G., Wadhwani, M., Lal, S., and Verma, T. Expert based recommendation system using community detection in online music streaming services. In *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1809–1813. IEEE, 2021.

Jannach, D., Lerche, L., and Zanker, M. Recommending based on implicit feedback. In *Social Information Access*, pages 510–569. Springer, 2018.

Javed, M. A., Younis, M. S., Latif, S., Qadir, J., and Baig, A. Community detection in networks: A multidisciplinary review. *J. Netw. Comput. Appl.*, 108:87–111, 2018.

Keerthi Gorripati, S. and Angadi, A. Visual based fashion clothes recommendation with convolutional neural networks. *International Journal of Information Systems & Management Science*, 1(1), 2018.

Kernighan, B. and Lin, S. An Efficient Heuristic Procedure for Partitioning Graphs. *The Bell Systems Technical Journal*, 49(2), 1970.

Kherad, M. and Bidgoly, A. J. Recommendation system using a deep learning and graph analysis approach. *Computational Intelligence*, 38(5):1859–1883, 2022.

Kleinberg, J. M. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

Lalwani, D., Somayajulu, D. V., and Krishna, P. R. A community driven social recommendation system. In *2015 IEEE International conference on big data (big data)*, pages 821–826. IEEE, 2015.

Lancichinetti, A. and Fortunato, S. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

Lancichinetti, A. and Fortunato, S. Limits of modularity maximization in community detection. *Physical review E*, 84(6):066122, 2011.

Langville, A. N. and Meyer, C. D. A survey of eigenvector methods for web information retrieval. *SIAM review*, 47(1):135–161, 2005.

Lattanzi, S. and Sivakumar, D. Affiliation networks. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 427–434, 2009.

Leskovec, J., Rajaraman, A., and Ullman, J. D. *Mining of Massive Datasets*. Cambridge University Press, 2014.

Linden, G., Smith, B., and York, J. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, 7(1):76–80, 2003.

Lloyd, S. P. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28:129–137, 1982.

Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32, 2015.

Ma, X., Lu, H., and Gan, Z. Improving recommendation accuracy by combining trust communities and collaborative filtering. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 1951–1954, 2014a.

Ma, X., Lu, H., Gan, Z., and Ma, Y. Improving recommendation accuracy with clustering-based social regularization. In *Web Technologies and Applications: 16th Asia-Pacific Web Conference,*

APWeb 2014, Changsha, China, September 5-7, 2014. Proceedings 16, pages 177–188. Springer, 2014b.

MacQueen, J. B. Some methods for classification and analysis of multivariate observations. In Cam, L. M. L. and Neyman, J., editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

Massa, P. and Avesani, P. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24, 2007.

Middleton, S. E., Shadbolt, N. R., and De Roure, D. C. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):54–88, 2004.

Mishra, N., Chaturvedi, S., Mishra, V., Srivastava, R., and Bargah, P. Solving sparsity problem in rating-based movie recommendation system. In *Computational Intelligence in Data Mining*, pages 111–117. Springer, 2017.

Morvan, A., Choromanski, K., Gouy-Pailler, C., and Atif, J. Graph sketching-based massive data clustering. *CoRR*, abs/1703.02375, 2017.

Murtagh, F. and Contreras, P. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

Newman, M. *Networks*. Oxford university press, 2018.

Newman, M. E. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006a.

Newman, M. E. Equivalence between modularity optimization and maximum likelihood methods for community detection. *Physical Review E*, 94(5):052315, 2016.

Newman, M. E. and Girvan, M. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

Newman, M. E. J. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

Newman, M. E. J. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006b.

Newman, M. E. J. *Networks: An Introduction.* Oxford University Press, 2010.

Oard, D. W., Kim, J., et al. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, volume 83. WoUongong, 1998.

O'Connor, M. and Herlocker, J. Clustering items for collaborative filtering. In *Proceedings of the ACM SIGIR workshop on recommender systems*, volume 128. UC Berkeley, 1999.

Paleti, L., Radha Krishna, P., and Murthy, J. Approaching the cold-start problem using community detection based alternating least square factorization in recommendation systems. *Evolutionary Intelligence*, 14:835–849, 2021.

Palla, G., Derényi, I., Farkas, I., and Vicsek, T. Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043):814–818, 2005.

Panteli, M., Piscopo, A., Harland, A., Tutcher, J., and Moss, F. M. Recommendation systems for news articles at the bbc. In *INRA@ RecSys*, pages 44–52, 2019.

Pasricha, H. and Solanki, S. A new approach for book recommendation using opinion leader mining. In *Emerging Research in Electronics, Computer Science and Technology: Proceedings of International Conference, ICERECT 2018*, pages 501–515. Springer, 2019.

Paul, P. and Aithal, P. Digital society: Its foundation and towards an interdisciplinary field. In *Proceedings of National Conference on Advances in Information Technology, Management, Social Sciences and Education*, pages 1–6, 2018.

Pazzani, M. J. and Billsus, D. Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer, 2007.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Pham, M. C., Cao, Y., Klamma, R., and Jarke, M. A clustering approach for collaborative filtering recommendation using social network analysis. *J. Univers. Comput. Sci.*, 17(4):583–604, 2011.

Pons, P. and Latapy, M. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26-28, 2005. Proceedings 20*, pages 284–293. Springer, 2005.

Pothen, A. Graph partitioning algorithms with applications to scientific computing. In Keyes, D. E., Sameh, A., and Venkatakrishnan, V., editors, *Parallel Numerical Algorithms*, pages 323–368. Kluwer, 1997.

Putra, A. A., Mahendra, R., Budi, I., and Munajat, Q. Two-steps graph-based collaborative filtering using user and item similarities: Case study of e-commerce recommender systems. In *2017 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6. IEEE, 2017.

Quan, T. K., Fuyuki, I., and Shinichi, H. Improving accuracy of recommender system by clustering items based on stability of user similarity. In *2006 International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*, pages 61–61. IEEE, 2006.

Raghavan, U. N., Albert, R., and Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

Reichardt, J. and Bornholdt, S. Statistical mechanics of community detection. *Physical review E*, 74(1):016110, 2006.

Ricci, F., Rokach, L., and Shapira, B. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.

Ricci, F., Rokach, L., and Shapira, B. *Recommender Systems Handbook*. Springer, 2015.

Rich, E. User modeling via stereotypes. *Cognitive science*, 3(4):329–354, 1979.

Roetzel, P. G. Information overload in the information age: a review of the literature from business administration, business psychology, and related disciplines with a bibliometric approach and framework development. *Business research*, 12(2):479–522, 2019.

Roser, M., Ritchie, H., and Ortiz-Ospina, E. Internet. *Our World in Data*, 2015.

Rosvall, M. and Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, 105(4):1118–1123, 2008.

Sahebi, S. and Cohen, W. W. Community-based recommendations: a solution to the cold start problem. In *Workshop on recommender systems and the social web, RSWEB*, volume 60, 2011.

Sanderson, M. and Croft, W. B. The history of information retrieval research. *Proceedings of the IEEE*, 100(Special Centennial Issue):1444–1451, 2012.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Application of dimensionality reduction in recommender system-a case study. Technical report, Minnesota Univ Minneapolis Dept of Computer Science, 2000.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, 2001.

Sarwar, B. M., Karypis, G., Konstan, J., and Riedl, J. Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering. In *Proceedings of the fifth international conference on computer and information technology*, volume 1, pages 291–324, 2002.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, 2002.

Sharma, C. and Bedi, P. Ccfrs–community based collaborative filtering recommender system. *Journal of Intelligent & Fuzzy Systems*, 32(4):2987–2995, 2017.

Sharma, L. and Gera, A. A survey of recommendation system: Research challenges. *International Journal of Engineering Trends and Technology (IJETT)*, 4(5):1989–1992, 2013.

Speier, C., Valacich, J. S., and Vessey, I. The influence of task interruption on individual decision making: An information overload perspective. *Decision sciences*, 30(2):337–360, 1999.

Su, X. and Khoshgoftaar, T. M. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

Tang, L. and Liu, H. *Community Detection and Mining in Social Media*, 2010.

Tasgin, M., Herdagdelen, A., and Bingol, H. Community detection in complex networks using genetic algorithms, 2007.

Traag, V. A., Waltman, L., and Van Eck, N. J. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

Tran, T. and Cohen, R. Hybrid recommender systems for electronic commerce. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, 2000.

Wang, M., Gong, M., Zheng, X., and Zhang, K. Modeling dynamic missingness of implicit feedback for recommendation. *Advances in neural information processing systems*, 31:6669, 2018.

Werner, A. Organizing music, organizing gender: algorithmic culture and spotify recommendations. *Popular Communication*, 18(1):78–90, 2020.

West, D. B. et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

Xie, J., Kelley, S., and Szymanski, B. K. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4):1–35, 2013.

Xie, J. and Szymanski, B. K. Labelrank: A stabilized label propagation algorithm for community detection in networks. In *2013 IEEE 2nd Network Science Workshop (NSW)*, pages 138–143. IEEE, 2013.

Xin, L., E, H., Song, J., Song, M., and Tong, J. Book recommendation based on community detection. In *Pervasive Computing and the Networked World: Joint International Conference, ICPCA/SWS 2013, Vina del Mar, Chile, December 5-7, 2013. Revised Selected Papers*, pages 364–373. Springer, 2014.

Xu, B., Bu, J., Chen, C., and Cai, D. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30, 2012.

Yang, C. C., Chen, H., and Hong, K. Visualization of large category map for internet browsing. *Decision support systems*, 35(1):89–102, 2003.

Yang, Z., Wu, B., Zheng, K., Wang, X., and Lei, L. A survey of collaborative filtering-based recommender systems for mobile internet applications. *IEEE Access*, 4:3273–3287, 2016.

Yildirim, H. and Krishnamoorthy, M. S. A random walk method for alleviating the sparsity problem in collaborative filtering. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 131–138, 2008.

Zeng, Y. and Liu, J. Community Detection from Signed Social Networks Using a Multi-objective Evolutionary Algorithm. In Handa, H., Ishibuchi, H., Ong, Y.-S., and Tan, K. C., editors, *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1*, pages 259–270, Cham. Springer International Publishing, 2015.

Zhang, K., Zhang, Z., Bian, K., Xu, J., and Gao, J. A personalized next-song recommendation system using community detection and markov model. In *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*, pages 118–123. IEEE, 2017.

Zhang, P. and Chartrand, G. *Introduction to graph theory*. Tata McGraw-Hill, 2006.

Zhang, S., Yao, L., Sun, A., and Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.

Zhang, Y. and Liu, B. *Web Mining and Social Networking: Techniques and Applications*. Springer, 2010.

Zhao, Y., Levina, E., and Zhu, J. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.

Zhou, T., Ren, J., Medo, M., and Zhang, Y.-C. Bipartite network projection and personal recommendation. *Physical Review E*, 76(4):720+, 2007.