Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Aplicação do Aprendizado por Transferência para Otimização de Sistemas de Recomendação Federados

Vinícius Brandão Araújo

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Metodologia e Técnicas da Computação

Leandro Balby Marinho
(Orientador)

Campina Grande, Paraíba, Brasil

A663a Araújo, Vinícius Brandão.
Aplicação do aprendizado por transferência para otimização de sistemas de recomendação federados / Vinícius Brandão Araújo. – Campina Grande, 2024.
92 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2024.
"Orientação: Prof. Dr. Leandro Balby Marinho".
Referências.

1. Aprendizado por Transferência. 2. Sistemas de Recomendação Federados. 3. Fatoração de Matrizes Federada (FMF). 4. Privacidade de Dados. 5. Redução de Comunicação. 6. Metodologia e Técnicas da Computação. I. Marinho, Leandro Balby. II. Título.

CDU 004.78(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM CIENCIA DA COMPUTACAO
Rua Aprígio Veloso, 882, Edifício Telmo Silva de Araújo, Bloco CG1, - Bairro Universitário, Campina Grande/PB, CEP 58429-900
Telefone: 2101-1122 - (83) 2101-1123 - (83) 2101-1124
Site: http://computacao.ufcg.edu.br - E-mail: secpg@computacao.ufcg.edu.br

FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

VINÍCIUS BRANDÃO ARAÚJO

APLICAÇÃO DO APRENDIZADO POR TRANSFERÊNCIA PARA OTIMIZAÇÃO DE SISTEMAS DE RECOMENDAÇÃO FEDERADOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de Mestre em Ciência da Computação.

Aprovada em: 04/09/2024

Prof. Dr. LEANDRO BALBY MARINHO, Orientador, UFCG

Prof. Dr. HERMAN MARTINS GOMES, Examinador Interno, UFCG

Prof. Dr. FREDERICO ARAUJO DURÃO, Examinador Externo, UFBA

Documento assinado eletronicamente por **LEANDRO BALBY MARINHO**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 06/09/2024, às 15:27, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Frederico Araújo Durão**, **Usuário Externo**, em 06/09/2024, às 15:48, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **HERMAN MARTINS GOMES**, **PROFESSOR 3 GRAU**, em 23/09/2024, às 16:53, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **4773694** e o código CRC **DA7CA107**.

Referência: Processo nº 23096.060358/2024-51 SEI nº 4773694

Resumo

Este trabalho investiga a aplicação do Aprendizado por Transferência na otimização de Sistemas de Recomendação Federados (SRFs), visando superar desafios como custos elevados de comunicação. Através de uma abordagem que combina técnicas de pré-treinamento e transferência de aprendizado com o paradigma de aprendizagem federada, o estudo propõe uma metodologia para melhorar a eficiência e eficácia dos SRFs.

A pesquisa foi motivada pela crescente necessidade de sistemas de recomendação que operem de forma eficiente em ambientes federados, preservando a privacidade dos dados do usuário enquanto fornecem recomendações personalizadas. Em resposta a esses desafios, o estudo foca na otimização do algoritmo de Fatoração de Matriz Federada (FMF), utilizando técnicas de aprendizado por transferência.

A metodologia empregada envolve o uso de conjuntos de dados reconhecidos, como o MovieLens e o Netflix Prize, para treinar modelos de recomendação. A pesquisa explora estratégias de transferência de conhecimento de sistemas de recomendação centralizados, ou seja, aqueles treinados centralmente, para a abordagem federada. Investigamos técnicas de *embeddings*, tais como PCA (Análise de Componentes Principais) e Word2Vec, em conjunto com o treinamento federado, para avaliar os diferentes embeddings gerados e utilizá-los no processo de treinamento federativo.

Os resultados obtidos demonstram a viabilidade das abordagens propostas, evidenciando reduções no custo de comunicação e no número de usuários necessários para alcançar a convergência. Além disso, um estudo de caso focado na conformidade com a Lei Geral de Proteção de Dados (LGPD) é apresentado, ressaltando a relevância prática das técnicas desenvolvidas para sistemas de recomendação em conformidade com regulamentações de proteção de dados.

Em suma, este trabalho contribui para a área de Sistemas de Recomendação Federados, demonstrando como a integração do Aprendizado por Transferência com a aprendizagem federada pode otimizar a performance dos SRFs. As implicações práticas deste estudo são relevantes para o desenvolvimento de sistemas de recomendação que garantam a privacidade dos dados do usuário.

Abstract

This study investigates the optimization of Federated Recommendation Systems (FRSs) by applying Transfer Learning techniques. Facing challenges posed by high communication costs and the necessity to preserve user data privacy, we propose an innovative methodology that leverages pre-training and knowledge transfer strategies. This research uses well-known datasets such as MovieLens and Netflix Prize to explore knowledge transfer from centralized recommendation systems to a federated approach. Techniques such as PCA (Principal Component Analysis) and Word2Vec are examined in conjunction with federated training to evaluate and utilize the generated embeddings in the federated training process. The findings demonstrate significant reductions in communication costs and the number of users required for effective model training without compromising recommendation accuracy. Additionally, a case study focusing on compliance with the General Data Protection Regulation (GDPR) highlights the practical relevance of the developed techniques for data protection-compliant recommendation systems. This work contributes to the field of Federated Recommendation Systems by showing how integrating Transfer Learning with federated learning can optimize FRS performance, with practical implications for developing privacy-preserving recommendation systems in the current technological landscape.

Agradecimentos

Gostaria de expressar minha mais sincera gratidão à minha namorada, Thayna Lemos, por seu apoio, paciência e incentivo ao longo de toda esta jornada. Agradeço também à minha família, cujo apoio foram fundamentais para que eu chegasse até aqui. Um agradecimento especial ao meu orientador, Leandro Balby, por sua orientação, conhecimento e dedicação, que foram essenciais para a realização deste trabalho. Também gostaria de agradecer à Ericsson pelo apoio financeiro.

Conteúdo

Lista de Símbolos

FM - *Fatoração de Matrizes*

FMF - *Fatoração de Matrizes Federada*

GDE - *Gradiente Descendente Estocástico*

RMSE - *Root Mean Square Error*

PCA - *Principal Component Analysis*

LGPD - *Lei Geral de Proteção de Dados*

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Motivação

Os avanços tecnológicos têm proporcionado uma significativa aproximação entre dispositivos móveis e computadores, sobretudo no que concerne a processamento e memória. A conjugação desses elementos com a praticidade de acesso à internet faz com que esses dispositivos se tornem indispensáveis para uma gama diversificada de finalidades.

Dados fornecidos pelo Instituto Brasileiro de Geografia e Estatística (IBGE) [1] indicam que, no Brasil, os dispositivos móveis são o principal meio de acesso à internet no ambiente doméstico, sendo empregados em 99,5% dos domicílios conectados à rede. Este não é um fenômeno restrito ao contexto brasileiro, em 2022 [2], observou-se que o número de usuários conectados à internet por meio de um dispositivo móvel estava próximo de 5 bilhões.

Com base nesses dados, é evidente que a comunicação acessível e portátil, viabilizada pelos dispositivos, é uma realidade consolidada. Dentro desse panorama, os usuários demandam, com cada vez mais ênfase, experiências digitais de alto padrão. Estas não se referem somente à eficiência operacional, mas à oferta de uma experiência de usuário (UX), caracterizada por interfaces intuitivas, carregamentos velozes, personalização detalhada, conteúdo relevante e design sofisticado. Em áreas distintas, como e-commerce, serviços de streaming e redes sociais, o objetivo é proporcionar interações que satisfaçam e ultrapassem as expectativas dos usuários. Tal perspectiva é corroborada por estudos como os da Harvard

[1] https://acesse.one/noticia-ibge

[2] https://www.statista.com/forecasts/1146312/mobile-internet-users-in-the-world

Enterprise Architecture [3] , que destacam a necessidade de entender e responder às demandas dos usuários para entregar experiências digitais satisfatórias.

Considerando o grande volume de usuários que navegam por diversos serviços online, as empresas encontram-se em uma intensa competição para atrair esses usuários e convertê-los em potenciais clientes. Uma das maiores complexidades reside na personalização da experiência do usuário. Em vez de adotar experiências genéricas, aplicadas indistintamente a todos os usuários da plataforma, as empresas estão explorando diferentes abordagens ou métodos para fornecer conteúdo e interações. Essas abordagens variam desde o fornecimento de uma experiência uniforme para todos até a adaptação das interações às necessidades e interesses individuais dos usuários. Nesse cenário de busca por personalização, as empresas têm adotado com crescente ênfase a implementação de Sistemas de Recomendação (SRs), visando otimizar essa experiência individualizada.

As ferramentas de recomendação auxiliam na filtragem de itens que podem variar de filmes, em plataformas de streaming, a produtos em comércio eletrônico, entre outros. Esta filtragem destaca os itens mais relevantes para cada usuário de forma automatizada e personalizada. Uma ferramenta eficaz desse tipo pode aumentar a lealdade do usuário à plataforma, oferecendo uma experiência adaptada às suas preferências individuais. Esse processo se fundamenta em características ou perfis de consumo, alinhando usuários com históricos parecidos para sugestões mais direcionadas.

O processo de identificação de usuários com padrões de comportamento semelhantes é conhecido como filtragem colaborativa [49]. Uma técnica proeminente nesse contexto é a fatoração de matrizes [31]. Ela desmembra a matriz de interações dos usuários em duas matrizes menores. Utilizando métodos de otimização, esses componentes são refinados para melhor representar os padrões originais de interação. Uma vez definidos, eles auxiliam na predição das preferências dos usuários em relação a itens ainda não explorados, possibilitando a entrega de sugestões personalizadas.

Para que esse processo seja efetivo, é imprescindível a coleta de dados dos usuários. Entretanto, recentemente, tem-se observado o surgimento de legislações, como a *General Data Protection Regulation* (GDPR)[4] na Europa e a Lei Geral de Proteção de Dados (LGPD) [5]

[3] https://enterprisearchitecture.harvard.edu/user-research-methods-and-recommendations

[4] https://gdpr-info.eu/

[5] https://www.planalto.gov.br/ccivil$_0$3/$_a$to2015 − 2018/2018/lei/l13709.htm

no Brasil, que objetivam a proteção dos dados desses usuários. A Lei nº 13.853, de 2019[6] inclusa na LGPD, confere ao usuário o poder de decisão sobre o compartilhamento de seus dados com a empresa provedora do serviço. Tal negativa pode resultar em um sistema de recomendação deficiente, visto que se torna complexo decifrar as suas preferências históricas.

Em face destes desafios, a comunidade de aprendizado de máquina tem dedicado esforços para o desenvolvimento de uma subárea denominada Aprendizagem Federada. Esse conceito, apresentado por Brendan McMahan et al. [41], propõe uma técnica de treinamento distribuído de modelos que mantém os dados dos usuários em seus respectivos dispositivos, garantindo assim a privacidade dos usuários.

Diversas abordagens têm sido propostas para enfrentar esses desafios no contexto dos Sistemas de Recomendação (SRs). Em 2019, Ammad-ud-din et al. [60] introduziram a Fatoração de Matriz Federada (FMF), uma técnica que permite a fatoração de matrizes de forma descentralizada, minimizando a necessidade de compartilhamento de dados sensíveis entre usuários e servidores. Nesse modelo, apenas as tabelas de itens, que são matrizes contendo informações sobre os itens avaliados, como características ou avaliações anteriores, são transferidas entre o usuário e o servidor. Isso mantém os dados de interação dos usuários protegidos localmente, melhorando a segurança dos dados ao reduzir a exposição de informações pessoais.

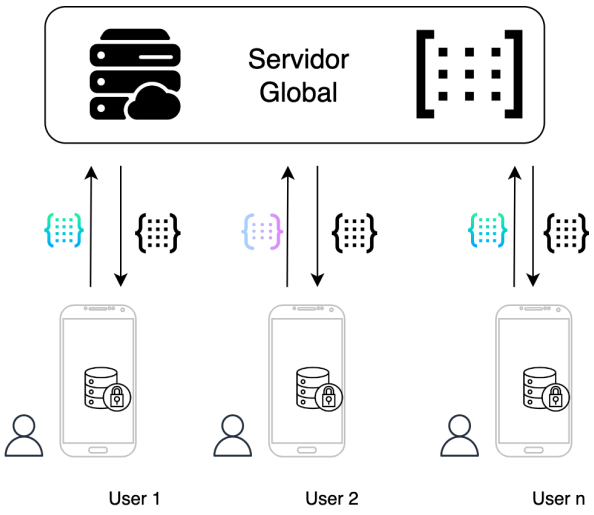

Figura 1.1: Exemplo de como funciona o FMF

Como ilustrado na Figura 1.1, a Fatoração de Matriz Federada (FMF), proposta por

[6]http://www.planalto.gov.br/ccivil$_03/_{Ato}2019-2022/2019/Lei/L13853.htmart1$

Ammad-ud-din et al. em 2019, é um método de aprendizado distribuído que mantém os dados sensíveis dos usuários diretamente em seus dispositivos, como telefones celulares. Este sistema realiza o treinamento em várias rodadas, onde em cada uma delas, uma amostra representativa dos usuários disponíveis é selecionada para contribuir no treinamento do modelo. Este processo visa alcançar a convergência, que é o ponto em que as previsões do modelo são maximamente precisas em relação aos dados reais observados nos dispositivos dos usuários. Destaca-se que, nesse esquema, apenas as matrizes de itens são compartilhadas com o servidor global, reduzindo a necessidade de transmissão de informações pessoais e sensíveis e, consequentemente, aumentando a segurança e a privacidade dos dados dos usuários.

Esta pesquisa é motivada pela busca de aprimoramentos no algoritmo de FMF. O objetivo é otimizar a convergência do algoritmo, reduzindo a quantidade de rodadas e usuários necessários para o treinamento desse sistema.

1.2 Definição do Problema

A aprendizagem federada, enfrenta desafios que limitam sua eficácia, especialmente em sistemas de recomendação. Conforme evidenciado por Tian Li et al. [36], um dos principais desafios é a quantidade de rodadas de comunicação necessárias para alcançar a convergência do modelo, agravada pelos altos custos de comunicação e pela necessidade de manutenção constante da conectividade entre os dispositivos dos usuários e o servidor central.

Além disso, a heterogeneidade dos dispositivos, que varia em termos de capacidade de processamento, memória e conectividade, pode influenciar diretamente o tempo necessário para que todos os dispositivos contribuam efetivamente para o modelo global. O modelo global, no contexto da aprendizagem federada, é um modelo central que é atualizado coletivamente por contribuições de vários dispositivos distribuídos. Esta questão é reforçada por Jie Ding et al. [15], que destacam como a diversidade de hardware afeta a uniformidade e a eficiência do processo de aprendizagem federada.

A capacidade dos modelos de generalizar para novos usuários ou itens pouco representados é outro desafio crucial, especialmente relevante em sistemas de recomendação, onde a variedade de itens e preferências dos usuários é extensa. Estudos como os de Farwa K.

Khan et al.[30] e Muhammad et al.[44], que exploram técnicas para otimizar a comunicação e a precisão, ainda enfrentam limitações quanto à eficácia da generalização em cenários práticos.

Portanto, nosso estudo se dedica a enfrentar esses desafios críticos: reduzir o número de rodadas de comunicação necessárias para a convergência e aprimorar a capacidade de generalização dos modelos em cenários de recomendação variados. Essas questões são barreiras para a implementação da aprendizagem federada em larga escala, exigindo o desenvolvimento de soluções que permitam sua aplicação eficiente em ambientes dinâmicos e diversos.

1.3 Objetivos

No domínio dos sistemas de recomendação, o trabalho de Costa et al. [14] destacou a prática de transferir *embeddings* de itens de grandes conjuntos de dados para sistemas que lidam com conjuntos menores. Utilizando um algoritmo de fatoração de matrizes treinado em um grande conjunto de dados publicamente acessível, os autores transferiram essas *embeddings* - ou seja, as preferências aprendidas - para um sistema baseado em fatoração de matrizes treinado em um conjunto de dados menor. Os resultados positivos obtidos por eles nos incentivaram a adotar uma abordagem semelhante.

Neste trabalho, exploramos a integração de estratégias de transferência de *embeddings* de itens em Sistemas de Recomendação Federados, inspirando-nos no estudo realizado por Costa et al. [14]. Nossa abordagem envolve um processo em três etapas:

1. **Treinamento Centralizado**: Inicialmente, treinamos um sistema de recomendação (SR) em um grande conjunto de dados disponível publicamente e alinhado com o conjunto de dados menor, garantindo que ambos pertençam ao mesmo domínio. Chamamos esse processo de treinamento centralizado porque os dados estão todos reunidos em um único local para o treinamento, ao contrário do treinamento federado, onde os dados permanecem distribuídos nos dispositivos locais.

2. **Mapeamento e Transferência**: Em seguida, aplicamos um processo de mapeamento e transferência para identificar quais itens treinados na etapa anterior estão presentes no conjunto de dados menor (conjunto alvo) e transferimos os pesos correspondentes

para a matriz de itens.

3. **Treinamento Federado**: Finalmente, realizamos o treinamento federado utilizando os *embeddings* transferidos, permitindo que um sistema de recomendação de escala menor beneficie-se do treinamento de larga escala previamente realizado. Esse benefício ocorre porque alguns *embeddings* já possuem pesos atualizados, podendo reduzir o número de rodadas necessárias para atingir a convergência e, consequentemente, diminuindo a comunicação entre os clientes e o servidor.

Com este estudo, buscamos avaliar a eficácia da transferência de *embeddings* no contexto de sistemas de recomendação federados, proporcionando insights sobre a viabilidade e os benefícios dessa abordagem.

Para orientar a pesquisa em alinhamento com o objetivo central, as seguintes questões de pesquisa (QP) foram delineadas:

- (QP1): De que maneira a transferência dos *embeddings* dos itens — obtidos a partir de treinamentos em Sistemas de Recomendação centralizados com dados reais — influencia e potencialmente reduz o custo de comunicação durante o treinamento em um Sistema de Recomendação Federado?

- (QP2): A transferência dos pesos derivados dos *embeddings* dos itens, originários de Sistemas de Recomendação centralizados treinados em dados sinteticos, pode aprimorar o custo de comunicação do treinamento?

- (QP3): A transferência dos pesos dos *embeddings* dos itens, obtidos de Sistemas de Recomendação centralizados, podem reduzir a quantidade de usuários necessários para o treinamento dos SRFs?

- (QP4): De que forma a seleção e transferência de uma amostra específica de *embeddings* de itens — previamente treinados em um sistema de recomendação centralizado — afetam a velocidade e a eficácia da convergência em um sistema de recomendação federado?

Ao abordar essas questões, pretendemos não apenas validar a viabilidade da transferência de *embeddings* no contexto da aprendizagem federada, mas também identificar práticas e

estratégias que possam ser amplamente aplicadas para melhorar a eficiência e a eficácia dos sistemas de recomendação federados.

1.4 Contribuições

As principais contribuições deste trabalho são elencadas abaixo:

- Propusemos e implementamos uma arquitetura para os sistemas de Recomendação Federado, permitindo sua operacionalização com um custo de comunicação notavelmente diminuído como mostramos no capitulo 5.
- Nossa abordagem promove a redução do número de usuários necessários para o treinamento dos SRFs, resultando numa otimização do processo de aprendizagem como demostrado no capitulo 5.
- Nossa abordagem garante um bom *trade-off* entre a acurácia e o custo de comunicação nos SRFs, um dilema prevalente nesta área de estudo.

1.5 Organização do Trabalho

A estrutura deste documento está organizada da seguinte forma:

No Capítulo 2, abordamos os fundamentos dos sistemas de recomendação com ênfase na fatoração de matrizes e exploramos técnicas de pré-treino como *PCA* e o *Word2Vec*. Destacamos a relevância da Aprendizagem Federada, incluindo suas variantes Local e a Reconstrução Federada (FedRecon). Finalizamos o capítulo introduzindo o algoritmo central da nossa proposta, o FMF com FedRecon, que une esses conceitos na otimização de Sistemas de Recomendação Federados, alinhando-se à transferência de aprendizagem para aprimorar a eficiência dos modelos propostos.

No Capítulo 3, conduzimos uma análise bibliográfica dos trabalhos mais relevantes que possuem correlação e influenciaram nossa pesquisa. Este tópico é subdividido em duas principais áreas:

- Sistemas de Recomendação Centralizados.

- Sistemas de Recomendação Federados.

No Capítulo 4, detalhamos a metodologia aplicada neste estudo. Aqui, especificamos a abordagem adotada, apresentando os dados utilizados e o pipeline experimental, que engloba desde a etapa de treinamento do sistema de recomendação centralizado, passando pelo mapeamento na transferência dos *embeddings*, até o treinamento federado.

No Capítulo 5, exibimos os resultados obtidos em consonância com as questões de pesquisa propostas. Analisamos os principais aspectos investigativos e discutimos os achados.

Concluindo, no Capítulo 6, apresentamos as conclusões deste trabalho e sugerimos direcionamentos para pesquisas futuras.

Capítulo 2

Fundamentação Teorica

Neste capítulo, apresentamos conceitos importantes para a compreensão do nosso trabalho, iniciando com a exploração de métodos tradicionais como Fatoração de Matrizes e Gradiente Descendente Estocástico, além de discutirmos sobre a métrica RMSE (*Root Mean Square Error*) para avaliação de desempenho desses sistemas. À medida que avançamos, examinamos estratégias de Pré-treino e Transferência de Aprendizado para otimizar a eficiência dos modelos, juntamente com a importância de *Embeddings*, destacando técnicas como *Word2Vec* e *PCA*. Damos continuidade ao tema abordando a Aprendizagem Federada Local e a Reconstrução Federada (FedRecon), que preparam o terreno para a introdução do algoritmo central da nossa proposta, o FMF com FedRecon.

2.1 Sistema de Recomendação

Sistemas de Recomendação são tipos de Sistemas de Informação que se dedicam a sugerir produtos, serviços ou informações relevantes a um usuário, baseado em diversos critérios e métricas. Estes sistemas têm sido efetivamente utilizados em diversos domínios como e-commerce, serviços de streaming de mídia, redes sociais e motores de busca, para citar alguns [37; 66; 23].

O principal objetivo dos Sistemas de Recomendação é preencher a lacuna informacional existente entre o vasto universo de opções disponíveis e a capacidade limitada do usuário em processar tais informações. Eles são projetados para modelar as preferências do usuário, fornecendo recomendações personalizadas que potencializam a satisfação do usuário, retenção

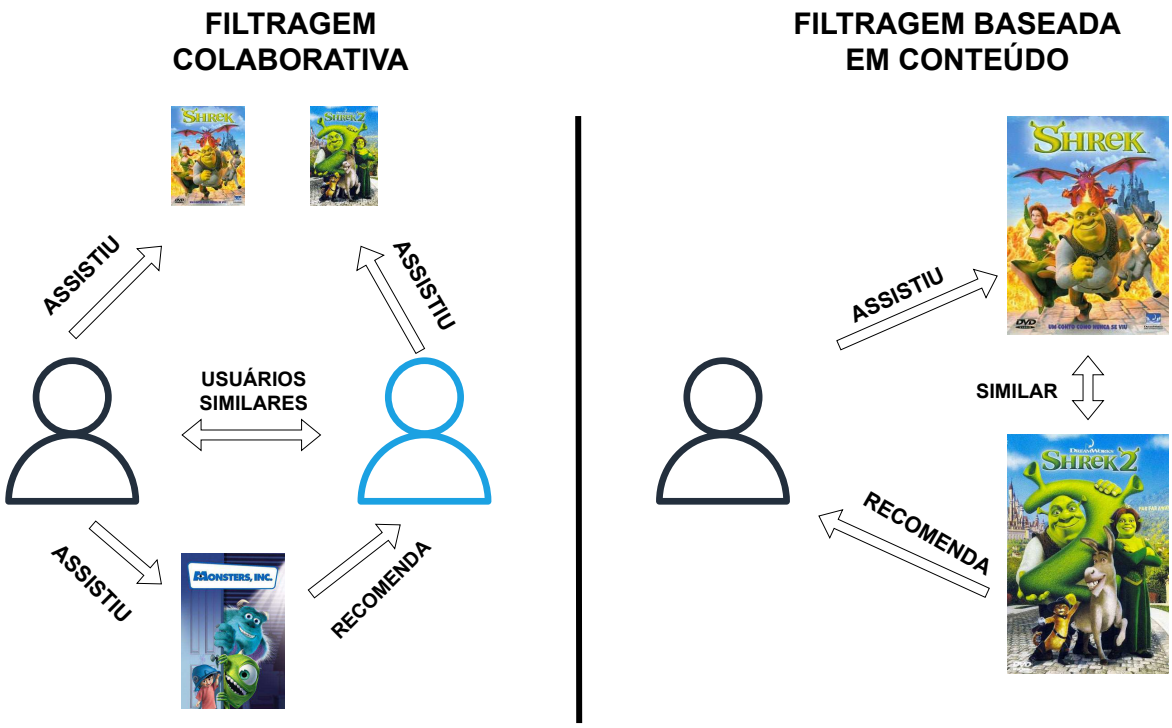

Figura 2.1: Tipos de Sistemas de Recomendação, Filtragem Colaborativa (a esquerda) e Baseados em Conteúdo (a direita)

e, em diversos contextos, monetização [24].

Esses sistemas podem ser predominantemente categorizados em três grupos: baseados em conteúdo, filtragem colaborativa e sistemas híbridos.

Sistemas de Recomendação Baseados em Conteúdo: Estes sistemas sugerem itens similares por meio de uma análise comparativa entre o conteúdo do item e o perfil do usuário. O conteúdo de cada item é caracterizado por descritores, tais como palavras-chave, categorias e metadados. Um exemplo de metadado pode ser o gênero do filme, como **animação** ou **comédia**. Por exemplo, na figura à direita, se o usuário **A** assistiu ao filme **Shrek 1**, que é descrito pelos gêneros **animação** e **comédia**, e o sistema identifica **Shrek 2** como possuindo gêneros semelhantes, então **Shrek 2** é recomendado ao usuário.

Sistemas de Recomendação baseados em Filtragem Colaborativa: Baseiam-se na ideia de que usuários com preferências semelhantes no passado tendem a continuar compartilhando preferências similares no futuro. Tais sistemas levam em conta o comportamento dos vários usuários, como avaliações de produtos e histórico de compras [33]. Por exemplo, na Figura 2.1 à esquerda, se o usuário **A** assistiu a *Shrek 1*, *Shrek 2* e *Monstros S.A.*, e o

usuário **B** viu *Shrek 1* e *Shrek 2*, o filme *Monstros S.A.* pode ser recomendado ao usuário B com base na sua similaridade de preferências.

Contudo, um desafio dos sistemas de filtragem colaborativa é o problema de *cold-start*, que torna difícil recomendar itens para novos usuários ou novos itens a usuários existentes devido à falta de interações no sistema [52].

Sistemas de Recomendação Híbridos: Estes sistemas combinam características das abordagens baseadas em conteúdo e filtragem colaborativa, buscando superar as limitações de cada uma e potencializar suas vantagens.

Tendo estabelecido a estrutura fundamental dos sistemas de recomendação, é fundamental discutir os detalhes técnicos que tornam essas recomendações precisas e eficazes. Uma das técnicas essenciais nesse contexto é a fatoração de matrizes [33], especialmente quando implementada usando gradiente descendente estocástico (GDE). Esta abordagem tem se mostrado particularmente eficaz em muitos sistemas de recomendação modernos. Além disso, para avaliar a precisão e confiabilidade dessas técnicas, métricas como o RMSE (*Root Mean Square Error*) são frequentemente utilizadas. Na próxima seção, vamos apresentar em mais detalhes a fatoração de matrizes via GDE e explorar como o RMSE pode ser usado para avaliar o desempenho de sistemas de recomendação.

2.1.1 Fatoração de Matrizes

A fatoração de matrizes (FM) é uma técnica adotada em sistemas de recomendação, originária de trabalhos na área de álgebra linear e processamento de sinais [33]. Nas últimas décadas, destacou-se especialmente na previsão de *ratings*. Os *ratings* referem-se às avaliações ou classificações que usuários atribuem a itens, como filmes ou produtos, geralmente expressas em uma escala numérica (por exemplo, de 1 a 5 estrelas), que indicam o quanto o usuário gostou ou apreciou o item [33].

Conceitos Básicos e História

A ideia central da FM é decompor uma matriz esparsa, devido à ausência de avaliações para muitos pares usuário-item, em matrizes menores que capturam características latentes de usuários e itens. Essa abordagem surgiu como uma alternativa para lidar com o desafio de

matrizes esparsas e grandes volumes de dados [33].

Seja $R \in R^{|U|\times|I|}$ uma matriz esparsa de *ratings*, onde U representa o conjunto de usuários e I o conjunto de itens. Uma entrada específica $r_{u,i}$ desta matriz indica a avaliação (nota) que o usuário u deu para o item i. A FM busca representar R como $\hat{R} = PQ^T$, onde $P \in R^{|U|\times d}$ e $Q \in R^{|I|\times d}$. Aqui, P e Q contêm as representações latentes, ou *embeddings*, de usuários e itens, respectivamente. Os *embeddings* são vetores de dimensão d que capturam propriedades e preferências implícitas dos usuários e características dos itens [33].

A predição de um *rating* é dada por:

$$\hat{r}_{u,i} = \mu + b_u + b_i + \langle p_u, q_i \rangle \tag{2.1}$$

Onde μ é a média global dos *ratings*, e b_u e b_i representam os viéses do usuário e do item, respectivamente. Esta formulação específica é comumente referida como FM com viés, devido à inclusão dos termos de viés [32].

Treinamento e Otimização

A otimização dos parâmetros é realizada minimizando uma função de perda, frequentemente o erro quadrático regularizado $L2$:

$$l(r_{u,i}, \hat{r}_{u,i}) = (r_{u,i} - \hat{r}_{u,i})^2 + \lambda(b_u^2 + b_i^2 + ||p_u||^2 + ||q_i||^2) \tag{2.2}$$

Aqui, λ é um parâmetro de regularização, crucial para prevenir o *overfitting* ao restringir a magnitude dos parâmetros. A importância da regularização pode ser observada em diversos trabalhos, que demonstram sua eficácia em melhorar a generalização do modelo em conjuntos de teste [45].

Para iniciar o processo de aprendizado, as matrizes de fatores latentes de usuários e itens P e Q são geralmente inicializadas com valores aleatórios de uma distribuição normal $\mathcal{N}(0, \sigma)$. Os parâmetros são então ajustados usando técnicas de otimização, como a GDE (Gradiente Descendente Estocástico), discutida detalhadamente na próxima seção.

2.1.2 Gradiente Descendente Estocástico

O Gradiente Descendente Estocástico (SGD, do inglês *Stochastic Gradient Descent*) é um método de otimização amplamente utilizado para treinar uma grande variedade de algoritmos

de aprendizado de máquina, notadamente redes neurais profundas [21]. Surgindo como uma variação do tradicional Gradiente Descendente (GD) [9], o SGD busca encontrar mínimos locais de uma função por meio de iterações. Enquanto o GD computa o gradiente utilizando todos os pontos de dados, exigindo uma passagem completa pelos dados, o SGD estima esse gradiente a partir de um subconjunto aleatório, que pode ser composto por um único ponto de dados ou um pequeno lote. Esse recurso faz com que o SGD seja especialmente útil em cenários de conjuntos de dados de grande escala, permitindo uma convergência mais rápida ao reduzir o custo computacional de cada iteração [65].

A aplicação do SGD em FM é notória. Neles, o SGD é empregado para otimizar os *embeddings* de usuários e itens, com o propósito de minimizar a discrepância entre as interações usuário-item observadas nos dados e aquelas previstas pelo modelo de FM. Essa otimização é feita atualizando iterativamente os *embeddings* na direção que atenua essa discrepância, guiado pelos cálculos do SGD.

De forma mais específica, para cada interação observada, o SGD avalia o gradiente da função de perda em relação aos *embeddings* de usuários e itens. Posteriormente, atualiza esses *embeddings* subtraindo o gradiente, que é multiplicado por uma taxa de aprendizado [50]. Este processo é repetido inúmeras vezes; cada ciclo completo pelo conjunto de dados é conhecido como época. Conforme o número de épocas aumenta, os *embeddings* são refinados para melhor representar as interações usuário-item, aumentando a precisão do modelo FM nas recomendações. Vale destacar que a taxa de aprendizado e outros hiperparâmetros são fundamentais para a eficácia do SGD, podendo influenciar a velocidade e a estabilidade da convergência [58].

Para garantir uma avaliação precisa do desempenho do modelo, é importante dispor de uma métrica confiável. Na seção seguinte, exploraremos o *Root Mean Square Error* (RMSE), uma métrica comumente utilizada para esse fim.

2.1.3 *Root Mean Square Error*

RMSE-*Root-Mean-Square Error* é uma métrica empregada para avaliar a precisão de modelos de previsão, conforme expresso pela Equação 2.3. Quantificando a média das diferenças ao quadrado entre as previsões de um modelo e os valores reais observados, seguida da extração da raiz quadrada desse valor médio. A elevação ao quadrado dos erros permite que

grandes discrepâncias sejam penalizadas de forma mais acentuada, tornando o RMSE sensível a grandes erros na previsão [27].

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.3}$$

Para ilustrar a aplicação do RMSE, considere o Sistema de Recomendação de filmes. Suponhamos que o sistema gerou as seguintes previsões de avaliações de filmes para um usuário:

- Filme A: 3.5
- Filme B: 4.2
- Filme C: 2.8

E os valores reais das classificações do usuário são:

- Filme A: 4.0
- Filme B: 3.5
- Filme C: 3.0

Agora, para calcular o RMSE, seguimos os seguintes passos:

Calculamos a diferença entre cada previsão e o valor real, e elevamos ao quadrado. Por exemplo, para o Filme A, temos:

$$(3.5 - 4.0)^2 = 0.25 \tag{2.4}$$

Repetimos isso para todos os filmes e calculamos a média desses valores. Neste cenário, temos:

$$\frac{(0.25 + 0.49 + 0.04)}{3} = 0.26 \tag{2.5}$$

Finalmente, extraímos a raiz quadrada deste valor médio para obter o RMSE. Assim, o RMSE para as nossas previsões é:

$$\sqrt{0.26} \approx 0.51 \tag{2.6}$$

O RMSE oferece uma percepção quantitativa da magnitude média dos erros de previsão. Valores menores de indicam um modelo de previsão mais preciso, especialmente no contexto de Sistemas de Recomendação. Um RMSE reduzido sugere que o sistema é eficiente ao estimar avaliações que os usuários atribuiriam aos itens, neste caso, filmes [16].

2.2 Pré-Treinamento e Transferência de Aprendizado na Aprendizagem Profunda

Introdução ao Pré-treinamento e Transferência de Aprendizado

A aprendizagem profunda tem se beneficiado significativamente das técnicas de pré-treinamento e transferência de aprendizado, particularmente em áreas como processamento de linguagem natural e visão computacional [47; 56]. A essência dessas abordagens reside na ideia de aproveitar o conhecimento adquirido em um domínio para facilitar o aprendizado em outro, muitas vezes distinto.

Mecanismos e Aplicações

O processo de pré-treinamento e transferência de aprendizado geralmente inicia com um modelo treinado em um vasto conjunto de dados. Inicialmente, um modelo é treinado em um conjunto de dados extenso e diversificado para capturar padrões gerais. Posteriormente, os pesos das camadas iniciais desse modelo, que têm a capacidade de identificar características genéricas de baixo nível, são transferidos para um novo modelo que será refinado para uma tarefa específica [63].

Para ilustrar esse conceito, consideremos o seguinte exemplo: um modelo é inicialmente treinado com o conjunto de dados *MovieLens 20M*, uma compilação extensiva contendo 20 milhões de avaliações de filmes, utilizada para capturar padrões gerais de preferências de usuários. Este pré-treinamento permite que o modelo aprenda representações úteis e características gerais das preferências dos usuários em relação aos filmes.

Após esse pré-treinamento, ocorre a transferência de aprendizado quando os pesos do modelo treinado no *MovieLens 20M* são transferidos para um novo modelo que será refinado para o conjunto de dados *MovieLens 1M*, que contém 1 milhão de avaliações. Neste ponto, o modelo utiliza as características previamente aprendidas como um ponto de partida, e o ajuste fino (*fine-tuning*) é realizado adicionando camadas completamente conectadas que serão especificamente treinadas para melhorar o desempenho no novo conjunto de dados menor.

Este processo de transferência de aprendizado demonstra como os conhecimentos adquiridos em um contexto amplo (*MovieLens 20M*) podem ser efetivamente aplicados e refinados para um contexto mais específico (*MovieLens 1M*), otimizando o modelo para melhor desempenho em tarefas específicas dentro do domínio de sistemas de recomendação. Assim, a transferência de aprendizado ocorre quando os pesos treinados no grande conjunto de dados são reutilizados e ajustados para melhorar a performance em um novo conjunto de dados menor, reduzindo o tempo e os recursos necessários para treinar o modelo desde o início.

Técnicas Alternativas e Inicialização

Além da transferência direta de pesos, há outros métodos para pré-treinamento. Alguns modelos aproveitam técnicas de redução de dimensionalidade, como a Análise de Componentes Principais (PCA) [55], e abordagens não supervisionadas, como o **Word2Vec** [42], para inicialização. Ambos os conceitos, que serão discutidos em detalhes nas seções subsequentes, oferecem perspectivas distintas e valiosas sobre como representar e extrair informações dos dados.

2.2.1 *Embeddings*

Os *embeddings* representam uma técnica essencial para a conversão de dados de alta dimensão em espaços vetoriais de menor dimensão, mantendo propriedades estruturais cruciais dos dados originais [22]. Em particular, no campo do Processamento de Linguagem Natural (PLN), têm demonstrado eficácia notável [42].

Tradicionalmente, a codificação "one-hot"era predominantemente utilizada para representar palavras, resultando em vetores esparsos com dimensão igual ao tamanho do vocabu-

lário. No entanto, essa representação possui limitações, principalmente sua inabilidade em capturar semelhanças semânticas e sintáticas entre palavras [20]. Como solução, *embeddings* de palavras representam cada termo como vetores densos em espaços vetoriais de dimensões reduzidas. Essa representação é concebida de maneira que palavras com contextos similares possuam embeddings próximos, refletindo semelhanças semânticas e sintáticas.

Diversos algoritmos foram propostos para aprender *embeddings* de palavras, incluindo Word2Vec [42], GloVe [46], e FastText [8]. A maioria desses algoritmos se baseia em prever palavras de acordo com seu contexto, e, durante esse processo, os *embeddings* são ajustados para maximizar a precisão dessas previsões.

Além do PLN, a metodologia de *embeddings* tem encontrado aplicações em diversas outras áreas, como em sistemas de recomendação. Na técnica de fatoração de matriz para Sistemas de Recomendação, a matriz de interações usuário-item é decomposta, resultando em *embeddings* para usuários e itens [33]. Esses *embeddings* capturam os interesses latentes dos usuários e as características intrínsecas dos itens, facilitando a realização de previsões precisas sobre interações futuras.

2.2.2 Análise de Componentes Principais

A Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*) [55] é um método estatístico que utiliza uma transformação ortogonal para transformar um conjunto de observações de variáveis potencialmente correlacionadas em um conjunto de valores de variáveis linearmente não correlacionadas, denominadas componentes principais. Este processo é realizado em sequência, com cada componente subsequente possuindo a máxima variância possível, sendo, simultaneamente, perpendicular aos componentes anteriores.

A motivação central por trás da PCA é revelar a estrutura subjacente dos dados de forma que represente uma variância máxima nos mesmos. Quando correlações fortes entre variáveis são identificadas, busca-se reduzir essas dimensões originais em componentes principais, que são menores em número mas que conservam a maior parte da informação crucial.

PCA tem aplicações de sucesso em vários domínios, tais como bioinformática, finanças, ciências sociais, entre outras. No domínio da aprendizagem de máquina, a PCA é frequentemente utilizada para análise e visualização de dados, pré-processamento de dados e engenharia de características.

Abaixo estão as etapas detalhadas de como o *PCA* é realizado:

1. Normalização de Dados: É uma etapa de pré-processamento para padronizar as variáveis de entrada em uma escala comum. Isso é necessário porque a há sensibilidade à escala das variáveis. A maneira mais comum de padronização é dimensionar as variáveis para que tenham média zero e desvio padrão unitário.

2. Calculando a Matriz de Covariância: A matriz de covariância é uma matriz quadrada que contém as covariâncias associadas a diferentes pares de variáveis. Os elementos da diagonal da matriz de covariância correspondem às variações das variáveis. Fora da diagonal, a matriz de covariância tem covariâncias simétricas.

3. Calculando os Autovalores e Autovetores da Matriz de Covariância: Esses cálculos são necessários para determinar os componentes principais da matriz de dados. Os autovetores da matriz de covariância são direções para os eixos onde há a máxima variação, e esses eixos são os componentes principais. Os autovalores são simplesmente a magnitude dessas quantidades de variação.

4. Ordenando os Autovalores: Os componentes principais são ordenados por seus autovalores correspondentes. O autovetor com o maior autovalor é considerado o primeiro componente principal. O autovetor com o segundo maior autovalor é considerado o segundo componente principal, e assim por diante.

5. Redução de Dimensão: Os componentes principais são todos ortogonais entre si. Eles são independentes e não correlacionados. Uma vez ordenados os autovalores e autovetores, podemos descartar os componentes principais com autovalores menores, pois eles são menos informativos. Consequentemente, reduzimos o número de variáveis e mantemos aqueles que contêm a maior parte da variação.

6. Projetando os Dados: Os dados originais são então projetados nos principais componentes, resultando no conjunto de dados transformado.

Em sistemas de recomendação, Marinho et al. [40] demonstraram que iniciar os *embeddings* de usuários ou itens da Fatoração de Matriz (MF) com PCA pode melhorar significativamente a explicabilidade do modelo, além de aumentar a precisão das recomendações.

O avanço das técnicas de representação não termina com a PCA. Uma técnica complementar e amplamente empregada, especialmente no contexto do Processamento de Linguagem Natural, é o Word2Vec. Esta técnica de *embedding*, que é detalhada na próxima subseção, foca em representar palavras em vetores de alta dimensão, permitindo a captura de semelhanças semânticas e sintáticas.

2.2.3 *Word2Vec*

O *Word2Vec* compreende um conjunto de modelos projetados para gerar *embeddings* de palavras. Estes modelos utilizam algoritmos de aprendizado supervisionado que empregam redes neurais rasas. Seu principal objetivo é treinar a rede para reconstruir contextos linguísticos das palavras. Como resultado, as palavras são representadas como vetores de alta dimensão (refletido no nome *word2vec*) e posicionadas no espaço vetorial de forma que palavras semanticamente semelhantes estejam próximas entre si.

A técnica *Word2Vec* foi apresentada inicialmente por Mikolov et al. [42]. Posteriormente, no mesmo ano, uma continuação, foi apresentada pelo mesmo grupo de pesquisa [43], introduzindo aprimoramentos ao modelo, permitindo-lhe representar frases além de palavras isoladas.

Existem dois algoritmos principais associados ao *Word2Vec*: *Skip-Gram* e *CBOW* (*Continuous Bag of Words*). No *CBOW*, a palavra alvo é prevista a partir do seu contexto, enquanto no *Skip-Gram*, o contexto é previsto a partir da palavra alvo. Em ambos, o "contexto"refere-se a uma janela de palavras adjacentes.

No domínio dos sistemas de recomendação, técnicas inspiradas no *Word2Vec*, como *item2vec* [6] e *prod2vec* [61], foram adotadas para gerar *embeddings* de itens. Porém, diferente destas abordagens, que não levam em conta a ordem dos itens e cuja janela de contexto engloba todo o conjunto de itens de um usuário, em nosso trabalho, preservamos o aspecto original do *Word2Vec*, considerando uma janela de vizinhança de tamanho 'w'. Esta escolha é fundamentada na premissa de que o contexto de um item pode ser influenciado por seus itens vizinhos, aproximando, assim, os que co-ocorrem no espaço de *embeddings*.

Uma observação relevante é a distinção entre os *embeddings* originados pelo *PCA* e pelo *Word2Vec*. Enquanto o *PCA* utiliza avaliações de itens como pesos que podem impactar os *embeddings* resultantes, o *Word2Vec* foca na co-ocorrência de itens dentro de uma janela de

contexto, independentemente de suas respectivas avaliações.

2.3 Aprendizagem Federada

Após explorar os conceitos de pré-treinamento e transferência de aprendizado, observamos a capacidade dessas técnicas de treinar modelos robustos em grandes conjuntos de dados e, subsequentemente, adaptá-los para tarefas específicas. Entretanto, surge a questão de como efetuar essa transferência quando os dados residem em diversos dispositivos e não em uma única fonte centralizada, considerando os desafios associados à privacidade e à eficiência na transferência de dados. A resposta a esta pergunta reside na aprendizagem federada, um paradigma que permite treinar modelos diretamente nos dispositivos distribuídos, sem a necessidade de centralizar os dados. Ao invés disso, apenas os parâmetros do modelo são transferidos e atualizados de forma agregada. Este modelo federado, em conjunto com a transferência de *embeddings*, pode ser a chave para a próxima geração de sistemas de recomendação eficazes, preservando a privacidade dos dados e melhorando a eficiência do treinamento.

Introdução à Aprendizagem Federada

Em ambientes de aprendizado de máquina tradicionais, os dados geralmente são centralizados em um único servidor ou local. No entanto, com a proliferação de dispositivos móveis e da Internet das Coisas (IoT), os modelos frequentemente residem nos próprios dispositivos. No modelo centralizado, embora os dados possam residir localmente, eles precisam ser enviados para um servidor central para treinamento. O paradigma da aprendizagem federada surge como uma resposta aos desafios apresentados por essa descentralização. A aprendizagem federada propõe a criação de um modelo de aprendizado compartilhado entre múltiplos dispositivos ou nós, sem necessariamente centralizar os dados.

Esta estratégia é crucial em cenários onde é impraticável ou indesejável transmitir dados devido a preocupações com privacidade, restrições de largura de banda ou problemas de conectividade.

Origens e Motivação

O conceito de aprendizagem federada foi introduzido por especialistas do Google em 2017. Em seu trabalho seminal, *Communication-Efficient Learning of Deep Networks from Decentralized Data*, McMahan et al. [41] detalharam a abordagem federada, evidenciando-a como uma solução promissora para desafios intrínsecos ao aprendizado descentralizado.

Funcionamento da Aprendizagem Federada

A essência da aprendizagem federada é descentralizar o treinamento de modelos. Em vez de centralizar os dados para treinar um modelo em um servidor, o próprio modelo é distribuído para os dispositivos que detêm os dados. Estes dispositivos, então, treinam o modelo com seus respectivos dados locais, retornando ao servidor apenas as atualizações ou os gradientes do modelo, e não os dados em si. Após receber atualizações de todos os dispositivos participantes, o servidor central combina essas atualizações para melhorar o modelo global.

A Figura 2.2 ilustra esse processo:

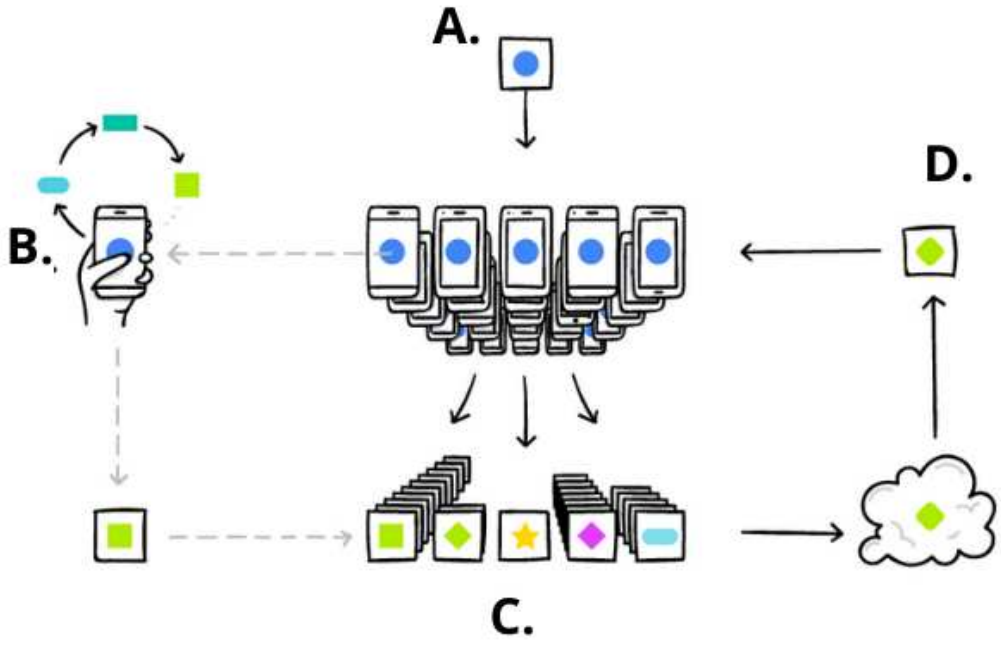

Figura 2.2: Esquema do processo de aprendizagem federada.

O ciclo de aprendizagem federada como representado na Figura 2.2, pode ser descrito nas seguintes etapas:

1. **Etapa A:** O servidor central inicia um modelo global e o distribui a todos os dispositivos participantes. Este modelo inicial é um ponto de partida comum para todos os dispositivos e serve como base para as atualizações locais.

2. **Etapa B:** Cada dispositivo participante atualiza o modelo global usando seus dados locais. Essa atualização é realizada através do ajuste dos pesos do modelo com base nas informações contidas nos dados específicos de cada dispositivo, aplicando algoritmos de otimização, como o Gradiente Descendente Estocástico (SGD).

3. **Etapa C:** Após o treinamento local, cada dispositivo envia suas atualizações de modelo (os gradientes ou os pesos atualizados) de volta para o servidor central. Notavelmente, apenas as atualizações dos parâmetros do modelo são enviadas, garantindo que os dados brutos dos usuários permaneçam locais e protegidos, preservando a privacidade.

4. **Etapa D:** O servidor central realiza a agregação das atualizações recebidas utilizando o algoritmo FedAvg (*Federated Averaging*). No FedAvg, o servidor calcula a média ponderada das atualizações de modelo recebidas de todos os dispositivos participantes, considerando o tamanho do conjunto de dados de cada dispositivo como peso. Esse processo assegura que as atualizações de dispositivos com mais dados tenham um impacto proporcionalmente maior na atualização do modelo global.

5. **Repetição do Ciclo:** O modelo global aprimorado é redistribuído aos dispositivos participantes, e o ciclo de atualização e agregação é repetido até que o modelo alcance uma convergência satisfatória. Esse método iterativo permite que o modelo global se beneficie das diversas atualizações dos dispositivos, adaptando-se continuamente com base nas novas informações coletadas localmente.

Existem diferentes métodos de agregação das atualizações, sendo a média ponderada das atualizações (FedAvg) uma das abordagens mais comuns, levando em conta a quantidade de dados de treinamento disponíveis em cada dispositivo.

2.3.1 Aprendizagem Federada Parcialmente Local e Reconstrução Federada

Após entender o funcionamento básico da aprendizagem federada, é importante destacar: a Aprendizagem Federada Parcialmente Local combinada com Reconstrução Federada. Essa abordagem, introduzida por Karan et al [57], propõe um método escalável que difere dos processos tradicionais de aprendizagem federada.

Aprendizagem Federada Local

Na aprendizagem federada tradicional, todos os parâmetros do modelo são atualizados localmente nos dispositivos dos usuários e, em seguida, agregados no servidor para atualizar o modelo global. Esta abordagem pode enfrentar desafios em cenários onde os dados dos usuários são heterogêneos, ou seja, quando há uma variação significativa nos tipos ou na distribuição dos dados entre os dispositivos dos usuários. Tal heterogeneidade pode comprometer a eficiência do modelo global, especialmente em aplicações sensíveis à privacidade, como sistemas de recomendação, onde parâmetros específicos do usuário, ao serem agregados, podem inadvertidamente revelar preferências pessoais sensíveis.

A Aprendizagem Federada Parcialmente Local propõe uma solução para isso, dividindo os parâmetros do modelo em globais e locais. Os parâmetros locais são específicos para cada dispositivo e não são compartilhados com o servidor, enquanto apenas os parâmetros globais são agregados. Essa abordagem reduz os riscos à privacidade, mantendo as preferências dos usuários confinadas aos seus dispositivos.

Reconstrução Federada (FedRecon)

A Reconstrução Federada (*FedRecon*) é uma abordagem específica da aprendizagem federada que promove uma integração mais eficiente entre o treinamento local e global. Em vez de manter os parâmetros locais constantes, o FedRecon utiliza um algoritmo de reconstrução para ajustar esses parâmetros em cada rodada de treinamento. Esse ajuste é baseado nos parâmetros globais atualizados e nos dados locais de cada dispositivo, conforme demonstrado pelo Algoritmo 1.

O processo de FedRecon inicia com a seleção aleatória de um subconjunto de dispositi-

Algorithm 1 Treinamento de Reconstrução Federada

Entrada: conjunto de parâmetros globais $\mathcal{G}$, conjunto de parâmetros locais $\mathcal{L}$, função de divisão do conjunto de dados S, algoritmo de reconstrução R, algoritmo de atualização do usuário U

O Servidor executa:

$g^{(0)} \leftarrow$ (inicializa $\mathcal{G}$)

for cada rodada t **do**

$\mathcal{S}^{(t)} \leftarrow$ (seleciona aleatoriamente m usuários)

for cada usuário $i \in \mathcal{S}^{(t)}$ **do**

$(\Delta_i^{(t)}, n_i) \leftarrow$ AtualizacaoUsuário$(i, g^{(t)})$

end for

$n = \sum_{i \in \mathcal{S}^{(t)}} n_i$

$g^{(t+1)} \leftarrow g^{(t)} + \eta_s \sum_{i \in \mathcal{S}^{(t)}} \frac{n_i}{n} \Delta_i^{(t)}$

end for

function ATUALIZACAOUSUÁRIO$(i, g^{(t)})$

$(\mathcal{D}_{i,s}, \mathcal{D}_{i,q}) \leftarrow S(\mathcal{D}_i)$

$l_i^{(t)} \leftarrow R(\mathcal{D}_{i,s}, \mathcal{L}, g^{(t)})$

$g_i^{(t)} \leftarrow U(\mathcal{D}_{i,q}, l_i^{(t)}, g^{(t)})$

$\Delta_i^{(t)} \leftarrow g_i^{(t)} - g^{(t)}$

$n_i \leftarrow |\mathcal{D}_{i,q}|$

retorna $(\Delta_i^{(t)}, n_i)$ para o servidor

end function

vos em cada rodada. Cada dispositivo selecionado divide seus dados em dois subconjuntos: um para reconstrução e outro para atualização do modelo. O algoritmo de reconstrução é então aplicado ao subconjunto de reconstrução, ajustando os parâmetros locais com base nos parâmetros globais atuais. Em seguida, o algoritmo de atualização utiliza esses parâmetros locais ajustados para refinar ainda mais os parâmetros globais com o subconjunto de atualização.

Esse processo de reconstrução permite que dispositivos que não participaram diretamente do treinamento do modelo global possam rapidamente adaptar seus parâmetros locais, personalizando o modelo para suas preferências específicas sem comprometer a privacidade. Além disso, essa abordagem melhora a eficiência computacional e a capacidade de escalabilidade em cenários *cross-device*, onde a disponibilidade dos dispositivos pode ser imprevisível. Em resumo, o FedRecon não só facilita uma melhor personalização do modelo em dispositivos individuais, mas também garante que o modelo global se beneficie de informações atualizadas de todos os dispositivos participantes.

Meta Aprendizagem e FedRecon

A essência da Reconstrução Federada está intimamente ligada aos princípios da Meta Aprendizagem [18]. A Meta Aprendizagem envolve a criação de modelos que podem adaptar-se rapidamente a novas tarefas com base em uma pequena quantidade de dados de treinamento. Na Reconstrução Federada, essa adaptabilidade é crucial, pois permite que os modelos ajustem rapidamente os parâmetros locais com base nos parâmetros globais atualizados e nos dados específicos do dispositivo, resultando em uma personalização eficiente e melhor desempenho em diversas condições de uso.

Portanto, o FedRecon não apenas aborda as preocupações com privacidade e escalabilidade na aprendizagem federada, mas também aproveita a flexibilidade da Meta Aprendizagem para melhorar a personalização do modelo em ambientes de aprendizagem federada. Esse avanço representa um passo significativo para a implementação de modelos que são ao mesmo tempo privados, escaláveis e altamente personalizáveis.

2.3.2 Sistema de Recomendação Federado

Os sistemas de recomendação federados, ou *Federated Recommendation Systems*, representam uma extensão do conceito de Aprendizagem Federada ao domínio dos sistemas de recomendação. Seu objetivo principal é proporcionar recomendações personalizadas aos usuários, treinando um modelo global distribuído por diversos dispositivos ou nós, com cada um mantendo seus dados locais.

Esses sistemas federados surgiram em resposta a questões de privacidade e eficiência no domínio dos sistemas de recomendação. Em sistemas de recomendação tradicionais, os dados de interação dos usuários costumam ser centralizados em um único servidor. Esta centralização, por um lado, gera preocupações significativas quanto à privacidade dos dados dos usuários. Por outro lado, o aumento contínuo no volume de dados torna cada vez mais complexo o processamento eficaz dessas informações em um ambiente centralizado. Dessa forma, os sistemas de recomendação federados não apenas abordam esses desafios, mas também oferecem vantagens adicionais, como redução de latência e maior personalização das recomendações, dado que os dados não precisam deixar o dispositivo do usuário.

Dentro dos sistemas de recomendação federados, uma técnica que tem recebido atenção é a Fatoração de Matrizes. Tradicionalmente empregada em sistemas de recomendação centralizados, a Fatoração de Matrizes busca decompôr a matriz de interação usuário-item em fatores latentes. No ambiente federado, essa técnica se adapta para garantir que a fatoração seja realizada sem comprometer a privacidade do usuário e mantendo os dados descentralizados. Na seção a seguir, exploramos a adaptação da Fatoração de Matrizes para um cenário federado, conhecida como *Fatoração de Matrizes Federada*.

2.3.3 Fatoração de Matrizes Federada

A Fatoração de Matrizes Federada (FMF), emergente no campo da aprendizagem federada, adapta-se ao contexto dos sistemas de recomendação. O conceito básico de FMF, semelhante ao método de fatoração de matrizes tradicional, busca modelar as interações dos usuários por meio da decomposição de matrizes em fatores latentes. A distinção crucial, no entanto, reside na execução: ao invés de centralizar os dados, a FMF executa operações em dispositivos locais, integrando as atualizações dos parâmetros de fatores latentes aprendidos individual-

mente em cada dispositivo através do processo de aprendizado federado.

Uma parte essencial desse processo é a transferência da matriz de itens entre os dispositivos. Ao compartilhar a matriz de itens, garantimos que todos os dispositivos participantes tenham uma representação consistente e atualizada dos itens. Isso é crucial porque permite que os dispositivos adaptem seus modelos locais utilizando informações globais, melhorando a qualidade das recomendações sem comprometer a privacidade dos dados dos usuários. A matriz de itens, ao ser transferida e ajustada continuamente, facilita a convergência do modelo federado, assegurando que as recomendações feitas sejam baseadas em um entendimento abrangente das preferências dos usuários, mesmo quando os dados permanecem distribuídos.

Alamgir et al. [1] elucida os desafios e avanços recentes na implementação de técnicas de fatoração federada. Estes avanços respondem a preocupações emergentes de privacidade e eficiência, onde a centralização de dados em sistemas de recomendação pode ser impraticável ou indesejável.

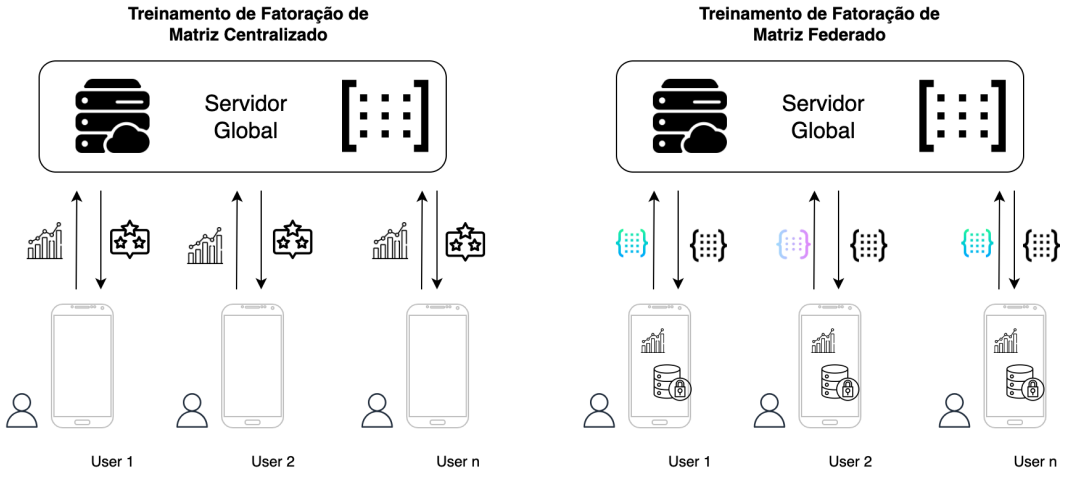

Figura 2.3: Contraste de arquiteturas entre fatorações de matriz em abordagens centralizadas e federadas.

Como ilustrado na Figura 2.3, sistemas centralizados coletam e processam dados no servidor, enquanto na FMF, os dados dos usuários são preservados em seus dispositivos, compartilhando somente informações cruciais para o modelo.

O procedimento de FMF pode ser delineado nas seguintes etapas:

1. O servidor inicialmente envia a matriz de itens I_q para dispositivos selecionados, es-

colhendo apenas a matriz de itens porque os *embeddings* de usuário são atualizados localmente em cada dispositivo e não necessitam ser compartilhados inicialmente.

2. Usando GDE (Gradiente Descendente Estocástico), cada dispositivo atualiza I_q e seus *embeddings* de usuário U_u baseando-se em uma função de perda predefinida, que mede a discrepância entre as interações observadas e as predições do modelo.

3. O servidor, em seguida, agrega as atualizações de I, que consistem nas modificações propostas aos *embeddings* dos itens por cada dispositivo, formando uma matriz de itens atualizada para a próxima iteração. Esta agregação reflete uma média ponderada das contribuições de cada dispositivo, atualizando assim a representação global dos itens.

Tal abordagem assegura a privacidade dos usuários ao evitar o compartilhamento direto de dados. Contudo, [10] salienta potenciais vulnerabilidades, como ataques adversários durante a agregação. Soluções, tais como agregações seguras e verificação de atualizações, são propostas para atenuar tais riscos.

O campo da FMF ainda está em evolução, com muitas questões em aberto relacionadas à otimização, privacidade, e escalabilidade. Em seções subsequentes, exploramos as implicações dessas técnicas nos sistemas de recomendação federados e discutimos suas potencialidades e limitações no cenário contemporâneo.

2.3.4 FMF com FedRecon

No contexto de sistemas de recomendação federados, o algoritmo Federated Reconstruction (FMF-FedRecon) oferece vantagens significativas em comparação com a tradicional Fatoração de Matrizes Federada (FMF). Enquanto a FMF foca na colaboração entre dispositivos para treinar um modelo global sem compartilhar dados sensíveis, o FedRecon leva essa abordagem mais além, introduzindo a reconstrução de parâmetros locais, o que traz várias melhorias:

1. **Personalização rápida para novos usuários**: O FedRecon permite a reconstrução de parâmetros locais em dispositivos de usuários, facilitando a personalização rápida do

modelo para novos usuários que não participaram do treinamento federado. Essa característica é especialmente valiosa em sistemas de recomendação, onde a capacidade de se adaptar rapidamente às preferências de novos usuários pode melhorar significativamente a experiência do usuário.

2. **Modelo agnóstico e escalável**: O FedRecon é projetado para ser agnóstico ao modelo, significando que pode ser aplicado a uma ampla gama de tarefas de aprendizado de máquina, incluindo sistemas de recomendação. Além disso, é escalável e compatível com treinamento em larga escala *cross-device*, uma vantagem importante ao lidar com um grande número de usuários e dispositivos em sistemas de recomendação.

3. **Eficiência de comunicação e robustez**: Ao focar na reconstrução de parâmetros locais e atualização apenas dos parâmetros globais, o FedRecon reduz o custo de comunicação e melhora a robustez à heterogeneidade dos dados dos usuários. Essas características são cruciais em sistemas de recomendação federados, onde a eficiência e a capacidade de lidar com diferentes distribuições de dados dos usuários são essenciais para o desempenho do sistema.

4. **Proteção contra vazamentos de privacidade**: O FedRecon, ao permitir que os parâmetros locais permaneçam nos dispositivos dos usuários e focando na reconstrução desses parâmetros quando necessário, oferece uma camada adicional de proteção contra vazamentos de privacidade. Isso é particularmente relevante em sistemas de recomendação, onde os dados dos usuários são sensíveis e a privacidade é uma grande preocupação.

5. **Amostragem de usuários e comunicação eficiente**: Um desafio comum na aprendizagem federada é a seleção de dispositivos para participação em cada rodada de treinamento, o que pode levar a alguns usuários não participarem regularmente do processo de treinamento. O FedRecon aborda isso permitindo que os dispositivos adaptem rapidamente seus parâmetros locais sempre que participam, garantindo que mesmo os usuários esporadicamente envolvidos possam beneficiar-se do modelo global atualizado. No entanto, a necessidade de transferir toda a matriz de *embeddings* de itens pode incorrer em gargalos de comunicação. Para mitigar isso, técnicas de compressão

e seleção de submatrizes de itens podem ser aplicadas, reduzindo o volume de dados transferidos e melhorando a eficiência da comunicação.

Em resumo, o FedRecon oferece um método avançado e adaptável para a implementação de sistemas de recomendação federados, superando limitações das abordagens tradicionais de FMF ao melhorar a personalização, escalabilidade, eficiência de comunicação e proteção da privacidade.

Embora a fatoração de matriz tenha sido tradicionalmente usada em configurações centralizadas, é especialmente relevante no aprendizado federado: as classificações do usuário podem residir em dispositivos de cliente separados e podemos querer aprender *embeddings* e recomendações para usuários e itens sem centralizar os dados. Uma vez que cada usuário tem uma incorporação de usuário correspondente, é natural que cada cliente armazene sua incorporação de usuário - isso é muito melhor escalável do que um servidor central que armazena todas as incorporações de usuário.

Uma proposta para trazer a fatoração de matriz para FL é a seguinte:

1. O servidor armazena e envia a matriz de itens aos clientes amostrados em cada rodada.

2. Os clientes atualizam a matriz de itens e sua incorporação de usuário pessoal usando SGD no objetivo acima.

3. As atualizações para a matriz de itens são agregadas no servidor, atualizando a cópia do servidor para a próxima rodada.

Esta abordagem é parcialmente local, isto é, alguns parâmetros do cliente nunca são agregados pelo servidor. Embora essa abordagem seja atraente, ela requer que os clientes mantenham o estado em todas as rodadas, ou seja, seus *embeddings* de usuário. Algoritmos federados com monitoração de estado são menos apropriados para configurações FL entre dispositivos: nessas configurações, o tamanho da população costuma ser muito maior do que o número de clientes que participam de cada rodada, e um cliente geralmente participa no máximo uma vez durante o processo de treinamento. Além de depender de estado que não pode ser inicializado, algoritmos *stateful* podem resultar em degradação do desempenho em ambientes de vários dispositivos devido ao estado ficando obsoleto quando os clientes raramente são amostrados. É importante ressaltar que na configuração de fatoração de matriz,

um algoritmo *stateful* leva a todos os clientes invisíveis perdendo *embeddings* de usuários treinados e, em treinamento em grande escala, a maioria dos usuários pode ser invisível.

Federated Reconstruction (FedRecon) é uma alternativa sem estado para a abordagem acima. A ideia principal é que, em vez de armazenar os *embeddings* do usuário em rodadas, os clientes reconstroem os *embeddings* do usuário quando necessário. Quando o FedRecon é aplicado à fatoração da matriz, o treinamento prossegue da seguinte forma:

1. O servidor armazena e envia a matriz de itens aos clientes amostrados em cada rodada.

2. Cada cliente congela a matriz de itens e treina sua incorporação de usuário utilizando um ou mais passos de SGD (reconstrução).

3. Cada cliente congela a incorporação de usuário e treina a matriz de itens usando uma ou mais etapas de SGD.

4. As atualizações para a matriz de itens são agregadas entre os usuários, atualizando a cópia do servidor para a próxima rodada.

Essa abordagem não exige que os clientes mantenham o estado entre as rodadas. Os autores também mostram no artigo que este método leva à reconstrução rápida de *embeddings* de usuário para clientes invisíveis, permitindo que a maioria dos clientes que não participam do treinamento tenham um modelo treinado, permitindo recomendações para esses clientes.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, buscamos aprofundar nosso entendimento sobre a evolução das abordagens adotadas em Sistemas de Recomendação, desde as tradicionais até as mais recentes baseadas em aprendizagem federada. Para esclarecer este panorama, discutiremos, inicialmente, estudos que exploram métodos de inicialização de *embeddings*, incluindo trabalhos que aplicam o *Word2Vec* e *PCA* em Sitemas de Recomendação em abordagens centralizadas. Em seguida, exploraremos os recentes avanços dos Sistemas de Recomendação Federados, analisando as principais abordagens e técnicas adotadas para superar os desafios inerentes a este domínio. A análise destes trabalhos proporcionará um sólido entendimento sobre o estado da arte e as tendências futuras neste campo de estudo.

3.1 Sistemas de Recomendação: Abordagens Centralizadas

No estágio inicial desta pesquisa, focamos nas abordagens de Sistemas de Recomendação treinados de maneira centralizada. Esta etapa visava compreender as abordagens estado da arte para a geração de *embeddings* de itens, que posteriormente seriam empregados na segunda etapa do estudo.

Nesse contexto, analisamos o estudo de Marinho et al. [40]. Esse trabalho destaca a importância da inicialização dos *embeddings* de usuários ou itens através do Método de Fatoração Matrizes (MF), empregando a Análise de Componentes Principais (*PCA*). Tal aborda-

gem beneficia o desempenho do modelo ao contribuir para sua explicabilidade, preservando a estrutura de vizinhança, e ao mesmo tempo, aprimorando a precisão das recomendações.

Costa et al. [14] propuseram uma aplicação distinta do modelo *Word2Vec* na inicialização dos *embeddings*. Ao contrário das abordagens tradicionais como *Item2Vec* e *Prod2Vec*, eles incorporam o princípio da janela de vizinhança do *Word2Vec*, posicionando itens co-ocorrentes mais próximos no espaço de *embedding*. Adicionalmente, a incorporação de carimbos de data/hora na ordenação dos itens proporciona recomendações mais contextualizadas e precisas.

3.2 Sistemas de Recomendação: Abordagens Federadas

Nos últimos anos, a adoção de aprendizado federado em sistemas de recomendação tem se destacado como uma solução promissora para preservar a privacidade dos usuários e otimizar a eficiência computacional. A seguir, discutimos diversas abordagens federadas propostas na literatura, destacando suas contribuições e limitações, culminando na nossa proposta de um sistema aprimorado com aprendizado por transferência.

3.2.1 Técnicas de Privacidade Diferencial

A privacidade e a segurança dos dados dos usuários são preocupações crescentes na área de sistemas de recomendação. Nesse contexto, a privacidade diferencial surge como uma ferramenta importante para proteger informações sensíveis dos usuários. A privacidade diferencial é uma abordagem que garante a proteção das informações sensíveis ao introduzir ruído nos dados de forma controlada, de modo que as informações individuais não possam ser inferidas. A seguir, apresentamos três trabalhos que exploram essa técnica em sistemas de recomendação federados.

Zhou et al. [67] propuseram um sistema de recomendação federado contextual que preserva a privacidade ao agregar estatísticas usando uma árvore de clusters e técnicas de privacidade diferencial. Este framework lida com grandes volumes de dados sem comprometer a privacidade dos usuários, utilizando mecanismos de privacidade diferencial como o mecanismo de *Laplace* e *Exponential* [17] para garantir a proteção das informações sensíveis.

Tan Li et al. [35] introduziram um framework que combina aprendizado federado e privacidade diferencial para sistemas de recomendação. A técnica *Upper Confidence Bound* (UCB) do algoritmo *multi-armed bandit* é utilizada para proteger a privacidade na agregação de estatísticas, equilibrando exploração e exploração na seleção de opções. Em vez de compartilhar dados brutos, apenas informações agregadas ou estatísticas criptografadas são transmitidas, preservando a privacidade dos indivíduos. Diferentemente da abordagem baseada em árvore de clusters de Zhou et al. [67], este trabalho adota o algoritmo *multi-armed bandit* com UCB, permitindo um balanceamento dinâmico entre exploração e aproveitamento na agregação de estatísticas com privacidade.

Chen Gao et al. [19] focaram em feedbacks implícitos e propuseram um framework para filtragem colaborativa local com privacidade diferencial. O método adota um mecanismo de proteção baseado em privacidade diferencial para ofuscar os dados de interação dos usuários antes de enviá-los ao servidor. O servidor então calcula as similaridades entre itens sem usar dados diretamente vinculados aos usuários e envia a matriz de similaridade de itens para os dispositivos dos usuários. As recomendações são feitas localmente, combinando essa matriz com os dados de comportamento armazenados localmente.

Embora Zhou et al.[67] e Tan Li et al.[35] utilizem privacidade diferencial para proteger os dados dos usuários, as abordagens diferem significativamente na maneira como implementam essa proteção. Zhou et al. empregam uma árvore de clusters para agregar estatísticas, o que é eficaz em lidar com grandes volumes de dados, mas pode ser menos flexível em termos de personalização. Em contraste, Tan Li et al. utilizam o algoritmo *multi-armed bandit* com UCB, que oferece um balanceamento dinâmico entre exploração e aproveitamento, adaptando-se melhor a cenários onde a dinâmica do usuário é mais complexa. Por outro lado, Chen Gao et al. [19] adotam uma abordagem mais focada em feedbacks implícitos e ofuscação dos dados, o que pode ser mais adequado para sistemas onde a privacidade é uma preocupação crítica, mas pode apresentar desafios em termos de complexidade computacional.

No entanto, enquanto a privacidade diferencial aborda a proteção direta dos dados dos usuários, outra linha de pesquisa foca em melhorar a eficiência e a personalização das recomendações. A seguir, exploramos métodos de fatoração de matrizes federada, que oferecem uma abordagem estruturada para o tratamento de dados em sistemas distribuídos.

3.2.2 Métodos de Fatoração de Matrizes Federada

As pesquisas nessa área exploram diferentes abordagens e desafios, como a proteção da privacidade, a eficiência de comunicação e a personalização das recomendações.

Gábor Danne et al. [26] comparou técnicas de aprendizado descentralizado, como *Gossip* e aprendizado federado, para fatoração de matrizes em sistemas de recomendação. Gossip é uma técnica onde cada nó da rede comunica periodicamente informações com nós vizinhos de forma aleatória, permitindo um consenso global sem um servidor central. Essa comparação destacou as vantagens e limitações de cada abordagem, mostrando que o aprendizado federado pode fornecer melhores garantias de privacidade.

Yujie Lin et al. [38] apresentou o MetaMF, um framework de fatoração de matrizes que utiliza aprendizado federado para realizar predições de classificações. O MetaMF inclui três componentes principais: um módulo de memória colaborativa (CM), um módulo de recomendação meta (MR) e um módulo de predição (RP). O módulo CM coleta informações colaborativas entre usuários para criar vetores colaborativos, enquanto o módulo MR gera *embeddings* privados de itens e modelos de predição. A estratégia de geração de aumento dimensional (RG) é aplicada para lidar com a alta dimensionalidade dos *embeddings* de itens, permitindo a criação de *embeddings* de alta dimensão a partir de matrizes de baixa dimensão. Os módulos CM e MR são executados no servidor central, enquanto o módulo RP é executado localmente nos dispositivos dos usuários.

Di Chai et al. [12] propôs um framework de fatoração de matrizes federada que utiliza criptografia homomórfica para proteger a privacidade dos usuários. Esta técnica permite operações sobre dados criptografados sem a necessidade de decifrá-los, garantindo que o servidor central não possa inferir os dados de preferência dos usuários a partir dos gradientes recebidos. Utilizando aprendizado federado, os modelos são treinados coletando informações de gradientes dos usuários em vez dos dados brutos. A inovação central é a integração da resposta randomizada e da criptografia homomórfica para proteger não apenas a privacidade dos valores e do modelo, mas também a existência dos dados dos usuários, ou seja, se um usuário participou ou não de determinada atividade.

O artigo de Shuai Wang et al. [62] propõe o FedMAvg, um novo algoritmo de fatoração de matrizes federada que combina minimização alternada e *model averaging* para melhorar a eficiência de comunicação em redes heterogêneas com dados não-i.i.d. Minimização alter-

nada é uma técnica de otimização que alterna entre a atualização de diferentes conjuntos de variáveis para encontrar a solução ideal. *Model averaging* , por sua vez, envolve a combinação de diferentes modelos treinados localmente para criar um modelo global mais robusto. Dados não-i.i.d (não independentemente e identicamente distribuídos) referem-se a conjuntos de dados onde as amostras não são independentes umas das outras e não seguem a mesma distribuição, o que é comum em cenários do mundo real onde os dados podem variar significativamente entre diferentes usuários ou dispositivos. O FedMAvg adota atualizações locais múltiplas e comunicação parcial entre clientes para reduzir o *overhead* de comunicação. A análise de convergência mostra que diminuir o número de atualizações locais pode reduzir a sensibilidade a dados não-i.i.d. Resultados experimentais em tarefas de clustering de dados e recomendação de itens demonstram a eficácia do FedMAvg em termos de desempenho e eficiência de comunicação.

Jia et al. [29] propuseram o FedMF), um algoritmo que utiliza a fatoração de matriz federada para personalizar recomendações em dispositivos móveis. A abordagem introduz correções de viés nos dados de avaliação dos usuários para ajustar com precisão as diferenças individuais e mitigar as avaliações anômalas, mantendo os dados de preferência dos usuários em seus dispositivos. Isso garante a privacidade e demonstra desempenho superior ao modelo FedMF tradicional, especialmente em termos de precisão de recomendação.

Os métodos de fatoração de matrizes federada discutidos variam amplamente em termos de suas abordagens para lidar com a privacidade e a eficiência da comunicação. Gábor Danne et al.[26] enfatizam a comparação entre aprendizado federado e descentralizado, sugerindo que o aprendizado federado pode oferecer melhores garantias de privacidade. Por outro lado, Yujie Lin et al.[38] introduzem o MetaMF, que combina aprendizado federado com meta-aprendizado para melhorar a precisão, mas pode envolver maior complexidade computacional o que difere de, Di Chai et al.[12] focam na privacidade utilizando criptografia homomórfica, oferecendo uma solução robusta, embora com custos computacionais mais elevados. Shuai Wang et al.[62], com o FedMAvg, propõem uma abordagem que otimiza a comunicação em redes heterogêneas, sendo ideal para cenários com dados não-i.i.d., mas pode sacrificar alguma precisão em comparação com outras abordagens. Jia et al. [29] abordam a personalização e a privacidade em dispositivos móveis, oferecendo uma solução que equilibra precisão e eficiência, mas pode enfrentar desafios em ambientes com grande

heterogeneidade de dados.

Embora a fatoração de matrizes federada tenha mostrado resultados promissores, ela não é a única abordagem para enfrentar os desafios dos sistemas de recomendação federados. Algumas técnicas combinam o melhor de diferentes mundos, explorando múltiplos aspectos como privacidade, eficiência e personalização. Na próxima seção, discutimos abordagens híbridas e outras técnicas que complementam os métodos discutidos anteriormente.

3.2.3 Abordagens Híbridas e Outras Técnicas

Além das técnicas de fatoração de matrizes e privacidade diferencial, a literatura apresenta uma variedade de outras abordagens para sistemas de recomendação federados. Estas abordagens exploram diferentes desafios e buscam soluções inovadoras para aprimorar a privacidade, a eficiência e a personalização das recomendações.

Fed-CARS (Waqar Ali et al. [2]): Framework de filtragem colaborativa baseado em aprendizado federado para sistemas de recomendação sensíveis ao contexto, que preserva a privacidade dos usuários. Em vez de transferir informações pessoais dos usuários para um servidor central, os dados são divididos em duas partes: dados pessoais e informações públicas compartilháveis. Utilizando um protocolo de colaboração definido pelo usuário (UDCP), os modelos locais são treinados nos dispositivos dos usuários com os dados pessoais e pesos de preferência fornecidos pelo servidor. O modelo global é treinado agregando os pesos enviados pelos usuários, garantindo a privacidade.

FedRecSys (Ben Tan et al. [59]): Permite a colaboração entre várias partes para treinar um modelo de recomendação sem comprometer a privacidade dos dados dos usuários. O sistema utiliza protocolos de computação segura baseados em criptografia homomórfica e compartilhamento secreto esta permite realizar operações matemáticas diretamente sobre dados criptografados sem precisar decifrá-los, enquanto compartilhamento secreto divide os dados em partes, distribuindo-as entre diferentes servidores para que apenas uma combinação específica possa revelar a informação original. O FedRecSys suporta algoritmos populares como fatoração de matriz e SVD. Implementado em um aplicativo de recomendação de conteúdo real, o FedRecSys resultou em uma melhoria significativa na taxa de cliques e no tempo médio de leitura su arquitetura do sistema inclui camadas de dados, algoritmos, serviços e interface, permitindo a gestão independente de dados e a atualização segura de

modelos.

FPPDM (Weiming Liu et al. [39]): Este framework de recomendação multidomínio com preservação de privacidade que utiliza modelagem probabilística de distribuição de preferências e *co-clustering* de compactação. Modelagem probabilística de distribuição de preferências envolve o uso de técnicas estatísticas para capturar e prever as preferências dos usuários com base em seus comportamentos passados.*Co-clustering* de compactação é uma técnica que agrupa simultaneamente usuários e itens com características semelhantes, reduzindo a dimensionalidade dos dados e melhorando a eficiência do processamento. O FPPDM consiste em dois componentes principais: um de modelagem de domínio local que captura distribuições de preferências de usuários e itens usando redes neurais, e um de agregação de servidor global que combina informações de usuários sobrepostos para compartilhamento de conhecimento. Além disso, o FPPDM++ introduz uma estratégia de *co-clustering* de compactação para agrupar usuários com características semelhantes, melhorando a precisão das recomendações. Experimentos em conjuntos de dados do Douban e da Amazon demonstram que o FPPDM/FPPDM++ supera os modelos de ponta em termos de desempenho, mantendo a privacidade dos dados dos usuários.

Método de otimização de carga útil (Farwa K. Khan et al. [30]): Aborda o desafio da crescente carga de comunicação em sistemas de recomendação federados. No contexto deste artigo, carga útil refere-se ao volume de dados que precisa ser transferido entre o servidor central e os dispositivos dos usuários durante as várias rodadas de treinamento de modelos. À medida que os modelos globais se tornam mais complexos, a quantidade de dados a serem transmitidos aumenta, o que pode causar latência e sobrecarga na rede. O método proposto aborda o desafio da crescente carga útil ao utilizar um modelo de *multi-armed bandit* para selecionar de forma inteligente apenas partes essenciais do modelo global a serem transmitidas. Isso reduz significativamente a quantidade de dados transferidos sem comprometer o desempenho das recomendações. Resultados experimentais demonstraram que o método pode alcançar uma redução de até 90% na carga útil, com uma perda de desempenho de apenas 4% a 8% em conjuntos de dados altamente esparsos. Esta abordagem permite a execução eficiente de FRS em cenários de produção, minimizando a carga de comunicação e mantendo a privacidade dos dados dos usuários.

FedeRank (Anelli et al. [3]) : UM sistema de recomendação federada que permite aos

usuários controlar os dados que compartilham com o servidor. FedeRank utiliza um modelo de fatoração pessoal em cada dispositivo, sincronizando o treinamento entre o servidor central e os clientes federados. Cada usuário pode optar por compartilhar uma quantidade mínima de dados sensíveis, mantendo o controle sobre suas informações pessoais. Experimentos mostraram que FedeRank oferece alta precisão nas recomendações mesmo com uma quantidade mínima de dados compartilhados, garantindo uma boa relação entre precisão e privacidade.

O artigo de Zeyu Cao et al. [11] apresenta um método para proteger a privacidade em sistemas de recomendação utilizando aprendizado federado vertical e *bandits* contextuais lineares. Aprendizado federado vertical permite que diferentes partes com diferentes atributos de um conjunto de dados colaborem sem compartilhar dados privados. *Bandits* contextuais lineares balanceiam exploração e exploração ao fazer recomendações com base em contextos adicionais. A abordagem utiliza o O3M (*Orthogonal Matrix-Based Mask Mechanism*), que aplica máscaras ortogonais às matrizes de dados para proteger as informações contextuais dos usuários. Os algoritmos LinUCB (*Linear Upper Confidence Bound*), que seleciona ações com base em um intervalo de confiança superior, e LinTS (Thompson Sampling), utilizando amostragem probabilística, foram adaptados para operar eficientemente neste ambiente federado vertical. Esses protocolos mantêm a qualidade das recomendações comparável aos algoritmos centralizados, com eficiência de tempo de execução satisfatória.

Chen et al. [13] propõe o framework SeSoRec para recomendações sociais seguras, que combina informações de plataformas sociais e de avaliação de usuários enquanto mantém a privacidade dos dados de ambas. Utilizando a técnica de *multiparty computation* (MPC), que permite que várias partes realizem cálculos conjuntos em seus dados sem revelar esses dados uns aos outros, juntamente com compartilhamento secreto, onde os dados são divididos em partes e distribuídos entre várias partes, esse sistema permite que as plataformas colaborem para melhorar a performance das recomendações sem compartilhar dados brutos. O protocolo SSMM (*Secret Sharing based Matrix Multiplication*) é central para a operação do SeSoRec, garantindo a segurança e a correção do processo de multiplicação de matrizes ao realizar operações matemáticas sobre dados compartilhados secretamente. Resultados experimentais com três conjuntos de dados reais demonstram a eficácia do SeSoRec, atingindo desempenho comparável aos modelos tradicionais de recomendação social, mas com

a segurança adicional proporcionada pela técnica de MPC.

Bichen Shi et al. [53] propõe um sistema de recomendação distribuído e assíncrono que utiliza aprendizagem por reforço profundo, baseado no modelo de vantagem ator-crítico assíncrono (A3C). A aprendizagem por reforço profundo combina redes neurais com algoritmos de aprendizagem por reforço para que os agentes aprendam a tomar decisões em um ambiente dinâmico. O modelo A3C é uma técnica onde vários agentes (atores) exploram o ambiente de forma assíncrona, enquanto um crítico avalia as ações tomadas para melhorar a política de decisão. O sistema, chamado DARES, combina abordagens de A3C e aprendizado federado, permitindo que os dados dos usuários permaneçam localmente em seus dispositivos. A arquitetura do sistema consiste em um modelo de recomendação local treinado nos dispositivos dos usuários e um modelo global treinado em um servidor central usando as atualizações de modelo calculadas nos dispositivos dos usuários. Avaliações usando conjuntos de dados bem conhecidos mostraram que, apesar de ser distribuído e assíncrono, o DARES pode alcançar um desempenho comparável e, em muitos casos, melhor do que os algoritmos atuais de ponta.

O artigo de Chengshuai Shi et al. [54] foi introduzido para integrar os princípios do aprendizado federado e a personalização ao problema dos *multi-armed bandits*. *Multi-armed bandits* é um problema clássico de otimização que balanceia exploração, onde um algoritmo deve decidir entre explorar novas opções ou as já conhecidas para maximizar a recompensa. O PF-MAB equilibra a generalização e a personalização ao formular um objetivo de aprendizado misto que combina informações locais e globais. O algoritmo PF-UCB (*Personalized Federated Upper Confidence Bound*) ajusta cuidadosamente o comprimento da exploração para alcançar esse equilíbrio e inclui uma análise teórica que demonstra que o PF-UCB pode alcançar um arrependimento (ıregret) de O(log(T)) independentemente do grau de personalização. Experimentos com dados sintéticos e reais corroboram a eficácia do algoritmo, destacando seu potencial em sistemas de recomendação onde a personalização e a privacidade são cruciais.

Khalil Muhammad et al. [44] propõe o FedFast, um framework que acelera o treinamento distribuído de sistemas de recomendação federada. FedFast utiliza duas técnicas: ActvSAMP, que seleciona os clientes que participam de cada rodada de treinamento, e ActvAGG, que combina os modelos treinados localmente de maneira eficiente, propagando as

atualizações para outros clientes. Essas técnicas reduzem os custos de comunicação e melhoram a precisão dos modelos desde os estágios iniciais do treinamento. FedFast é uma extensão do FedAvg (*Federated Averaging*), que é um algoritmo comum em aprendizado federado onde um modelo global é treinado iterativamente por meio da agregação das atualizações de modelos locais treinados em dispositivos dos usuários. Experimentos em diversos conjuntos de dados demonstram que o FedFast supera o método FedAvg em termos de velocidade de convergência e qualidade das recomendações, fornecendo modelos mais precisos com menor esforço de comunicação.

A abordagem do *Federated Reconstruction (FedRECON)* [57] combinou o treinamento federado de parâmetros globais com a reconstrução de parâmetros locais, permitindo a personalização rápida sem comunicação adicional. Este método é model-agnostic, adequado para grandes escalas de treinamento e inferência em configurações de dispositivos cruzados. FedRECON foi validado em tarefas de filtragem colaborativa e previsão de próxima palavra, mostrando desempenho superior em comparação com abordagens centralizadas e federadas tradicionais. A abordagem foi implementada em larga escala em um aplicativo de teclado móvel, aumentando a taxa de clique em 29,3% para recomendações de expressões.

As abordagens mencionadas nesta seção abrangem uma variedade de técnicas e enfoques para enfrentar os desafios dos sistemas de recomendação federados.Destacando algumas comparações:

- *Fed-CARS* e *FedRecSys*: Ambos focam na colaboração entre múltiplas partes e na proteção da privacidade dos usuários, mas o fazem utilizando diferentes protocolos e técnicas de computação segura. Enquanto o *Fed-CARS* adota um protocolo de colaboração definido pelo usuário para treinar modelos localmente nos dispositivos dos usuários, *FedRecSys* utiliza criptografia homomórfica e compartilhamento secreto para proteger os dados, permitindo maior segurança em ambientes altamente sensíveis.

- *FPPDM*: Essa abordagem se distingue por combinar modelagem probabilística e *co-clustering* para lidar com recomendações em múltiplos domínios, oferecendo uma solução eficaz para reduzir a carga de comunicação. Ao contrário de *Fed-CARS* e *FedRecSys*, que se concentram principalmente na privacidade, *FPPDM* equilibra eficiência e precisão, tornando-se ideal para cenários onde a comunicação eficiente é

crítica.

- *FPL* e *FedeRank*: Ambas as abordagens priorizam o controle do usuário sobre seus dados, permitindo que cada usuário decida quais informações compartilhar. *FedeRank* se destaca ao permitir um controle granular dos dados compartilhados, garantindo alta precisão nas recomendações com uma quantidade mínima de dados sensíveis compartilhados, o que pode ser mais flexível em comparação com as abordagens mais rígidas de *Fed-CARS* e *FedRecSys*.

- Zeyu Cao et al. [11]: Propõem um método inovador que combina aprendizado federado vertical com *bandits* contextuais, uma abordagem única em comparação com as outras, que geralmente se concentram em abordagens horizontais. Isso é especialmente útil em cenários onde diferentes partes possuem diferentes atributos de dados, oferecendo uma maneira eficaz de balancear exploração e exploração enquanto protege a privacidade.

- *SeSoRec*: Essa abordagem se diferencia por combinar informações de plataformas sociais e avaliações, com um foco mais específico na preservação da privacidade em ambientes sociais. Comparado a outras abordagens que tratam principalmente de dados transacionais ou de uso, *SeSoRec* é mais adequado para ambientes onde as interações sociais são uma parte crítica da experiência de recomendação.

- *DARES*: Utiliza *aprendizagem por reforço profundo* para criar um sistema de recomendação distribuído e assíncrono, o que é único entre as abordagens discutidas. Enquanto a maioria das abordagens se concentra em técnicas mais tradicionais de aprendizado de máquina, *DARES* explora a flexibilidade e o poder do aprendizado por reforço, tornando-o mais adaptável a ambientes dinâmicos.

- *PF-MAB*: Integra aprendizado federado e personalização no problema dos *multi-armed bandits*, destacando-se por sua capacidade de balancear generalização e personalização de forma eficaz. Isso contrasta com abordagens como *Fed-CARS* e *FedRecSys*, que focam mais em proteção de privacidade e menos em personalização dinâmica.

- *FedFast* e *FedRECON*: Ambas as abordagens se concentram em acelerar o treinamento distribuído, mas o fazem de maneiras diferentes. *FedFast* utiliza técnicas como Actv-

SAMP e ActvAGG para selecionar clientes e combinar modelos de forma eficiente, enquanto *FedRECON* combina treinamento federado com reconstrução local de parâmetros, permitindo personalização rápida sem comunicação adicional. *FedRECON*, em particular, se destaca por ser model-agnostic e adequado para grandes escalas de treinamento.

Esses trabalhos, em conjunto, demonstram o crescente interesse e os avanços na área de sistemas de recomendação federados, cada um abordando os desafios de privacidade, eficiência e personalização de diferentes maneiras. A diversidade de abordagens, desde aquelas que priorizam a segurança máxima dos dados até as que focam na eficiência de comunicação e personalização, evidencia o dinamismo do campo e abre caminho para futuras pesquisas que possam integrar o melhor de cada técnica, buscando soluções ainda mais eficientes e eficazes para os desafios da recomendação em um mundo cada vez mais conectado e preocupado com a privacidade."

As abordagens híbridas e técnicas avançadas discutidas até agora demonstram a diversidade e complexidade das soluções propostas na literatura. Para entender melhor o panorama atual e as direções futuras da pesquisa, a seguir, sintetizamos as principais tendências emergentes no campo dos sistemas de recomendação federados.

3.2.4 Síntese e Tendências

A revisão da literatura revela várias tendências importantes no campo dos sistemas de recomendação federados:

1. Foco crescente na privacidade e segurança dos dados dos usuários, com a adoção de técnicas como privacidade diferencial e criptografia homomórfica.

2. Busca por eficiência computacional e de comunicação em ambientes distribuídos, através de técnicas como minimização alternada e *model averaging*.

3. Desenvolvimento de técnicas para personalização e controle do usuário, permitindo que os usuários decidam quais dados compartilhar.

4. Integração de abordagens avançadas de aprendizado de máquina, como aprendizado por reforço e bandits contextuais, para melhorar a qualidade das recomendações.

5. Exploração de abordagens híbridas que combinam múltiplas técnicas para abordar desafios complexos em sistemas de recomendação federados.

Apesar dos avanços notáveis e das tendências emergentes identificadas, a análise revela áreas onde a pesquisa ainda pode evoluir significativamente. A próxima seção explora essas lacunas e apresenta nossa contribuição proposta, que visa superar algumas dessas limitações e avançar o estado da arte em sistemas de recomendação federados.

3.2.5 Lacunas de Pesquisa e Nossa Contribuição

Apesar dos avanços significativos, identificamos algumas lacunas importantes na literatura atual:

1. Limitada exploração do aprendizado por transferência em sistemas de recomendação federados, especialmente para lidar com a heterogeneidade dos dados entre usuários.

2. Necessidade de soluções mais eficientes para o problema de *cold-start* em ambientes federados, onde novos usuários ou itens têm dados limitados.

3. Falta de abordagens que equilibrem efetivamente a personalização local com a generalização global, mantendo a privacidade dos usuários.

4. Escassez de estudos sobre a escalabilidade e robustez de sistemas de recomendação federados em cenários do mundo real com grandes volumes de dados e usuários.

Nossa pesquisa visa abordar essas lacunas através da adoção e adaptação de aprendizado por transferência para aprimorar a eficiência desses sistemas. Propomos uma abordagem que explora a pré-inicialização de *embeddings* de itens, utilizando conhecimento prévio adquirido de grandes conjuntos de dados centralizados, para acelerar o processo de treinamento federado e, simultaneamente, melhorar a precisão das recomendações.

Esta abordagem não apenas aborda as preocupações com a privacidade dos dados ao minimizar a necessidade de comunicação entre o usuário e o servidor, mas também oferece uma solução prática para o problema de *cold-start* e a heterogeneidade dos dados nos dispositivos dos usuários por meio da reconstrução federada. Ao incorporar a técnica de aprendizado por transferência, propomos um sistema de recomendação federado mais adaptável, capaz

de oferecer recomendações personalizadas de qualidade com menor custo computacional e de comunicação.

Capítulo 4

Metodologia

Neste capítulo, apresentamos uma visão detalhada da metodologia empregada na construção de nossa abordagem. Delineamos as diversas fases do processo, iniciando com o treinamento centralizado, seguido pelo processo de mapeamento de dados e a transferência de *embeddings* treinados, importante para a integração dos diferentes conjuntos de dados. Em seguida, passamos para a implementação do procedimento de treinamento federado, utilizando o algoritmo de recomendação *FMF FedRecon*. Por fim, detalhamos a nossa escolha da arquitetura federada centralizada, que foi selecionada por sua eficácia em permitir controle sobre a matriz de itens e o processo de agregação para todos os usuários.

4.1 Conjuntos de Dados

Em nossa pesquisa, cinco conjuntos de dados foram utilizados, visando abordagens distintas de sistemas de recomendação. Os conjuntos de dados volumosos — MovieLens 1B[1], MovieLens 20M[2] e Netflix[3] — serviram para treinar o sistema centralizado, aproveitando sua ampla gama de avaliações de usuários. A Figura 4.1 demonstra a aplicação desses conjuntos no treinamento e os detalhes da transferência de aprendizado são explicados nas seções seguintes. Já os conjuntos MovieLens 1M[4] e MovieLens 100K[5], de menor escala, foram

[1] https://grouplens.org/datasets/movielens/movielens-1b/

[2] https://grouplens.org/datasets/movielens/20m/

[3] https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

[4] https://grouplens.org/datasets/movielens/1m/

[5] https://grouplens.org/datasets/movielens/100k/

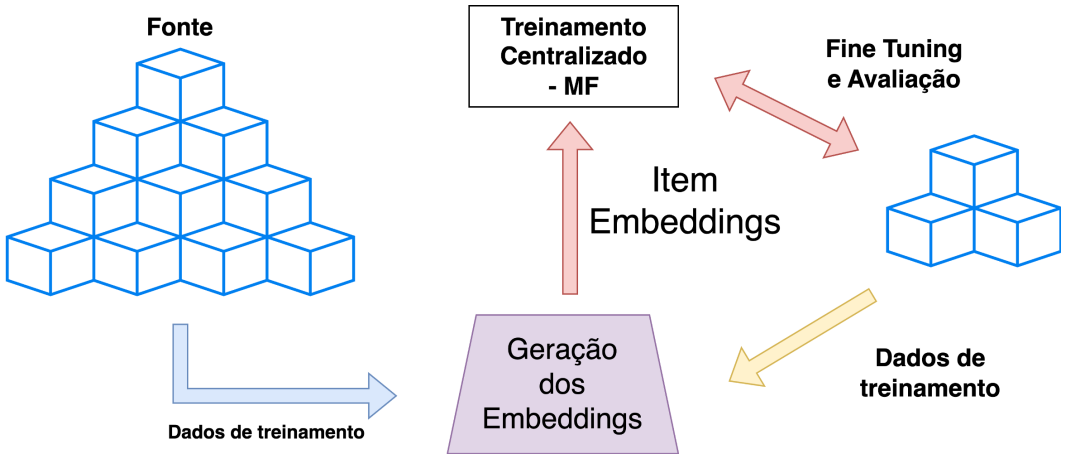

Figura 4.1: Transferência de conhecimento de um modelo centralizado treinado com amplos conjuntos de dados para um sistema federado utilizando conjuntos menores. Imagem adaptada de Costa et al [14].

utilizados para o treinamento do sistema federado.

4.1.1 MovieLens

Proveniente do GroupLens, laboratório de pesquisa da Universidade de Minnesota, os conjuntos de dados do MovieLens têm sido instrumentais para pesquisadores da área de sistemas de recomendação [25]. Estes datasets são compostos por avaliações de filmes feitas por usuários e têm variações em volume para adequar-se a distintas necessidades investigativas.

- **MovieLens 100k:** Ampla referência em estudos da área, este conjunto possui 100.000 avaliações de cerca de 1.682 filmes realizadas por 943 usuários. Inclui ainda dados demográficos dos usuários e informações dos filmes, permitindo uma diversidade de experimentações.

- **MovieLens 1M:** Expandindo o escopo do conjunto anterior, esta versão contém 1 milhão de avaliações de aproximadamente 4.000 filmes feitas por 6.000 usuários, mantendo informações demográficas e detalhes dos filmes.

- **MovieLens 20M:** Este dataset extenso traz 20 milhões de avaliações oriundas das interações de 138.000 usuários com cerca de 27.000 filmes, servindo como um recurso valioso para a avaliação de algoritmos em larga escala.

Os datasets do MovieLens têm fortalecido e direcionado avanços significativos na área de sistemas de recomendação, oferecendo um terreno fértil para avaliação e experimentação.

4.1.2 MovieLens 1B

Belletti et al. [7] desenvolveram o conjunto de dados MovieLens 1B, expandindo o MovieLens 20M com o uso da Teoria do Grafo de Kronecker [34] em matrizes de incidência usuário/item. Este processo permitiu a geração de interações sintéticas usuário-filme, simulando de maneira precisa padrões de engajamento e popularidade observados em ambientes reais. O resultado foi um conjunto de dados que melhor reflete as condições encontradas em sistemas de recomendação industriais, proporcionando uma ferramenta útil tanto para a pesquisa acadêmica quanto para o aprimoramento de algoritmos de recomendação em larga escala.

Possuindo mais de 1 bilhão de interações, o MovieLens 1B retrata o *feedback* implícito dos usuários, indicando se assistiram ou não determinado filme. Devido à sua vastidão, que demanda cerca de 400 GB de RAM, adotamos uma amostragem desses dados para o treinamento do sistema centralizado. Este processo de amostragem, descrito adiante, visa tornar nosso pipeline independente de conjuntos de dados reais e otimizar o processamento.

4.1.3 Netflix Prize

A empresa Netflix lançou o conjunto de dados Netflix Prize para um desafio competitivo. Composto por avaliações anônimas de filmes e suas interações, ele serve como uma fonte de *feedback* explícito dos usuários. As avaliações variam em uma escala de 1 a 5.

Dado o volume substancial do Netflix Prize, utilizamos uma amostra focada em filmes que também estão presentes no MovieLens 1M. Para isso, um processo de mapeamento foi realizado, já que os identificadores dos filmes nos dois conjuntos de dados não são os mesmos. Detalhes desse mapeamento são elucidados na seção 4.4.

4.2 Pipeline do Experimento

Durante esta pesquisa, desenvolvemos um pipeline para a execução do experimento. A abordagem consiste em três etapas fundamentais: o treinamento de um sistema de recomendação centralizado em um vasto conjunto de dados, o mapeamento dos dados que serão utilizados no processo federado e, finalmente, o treinamento do sistema de recomendação federado.

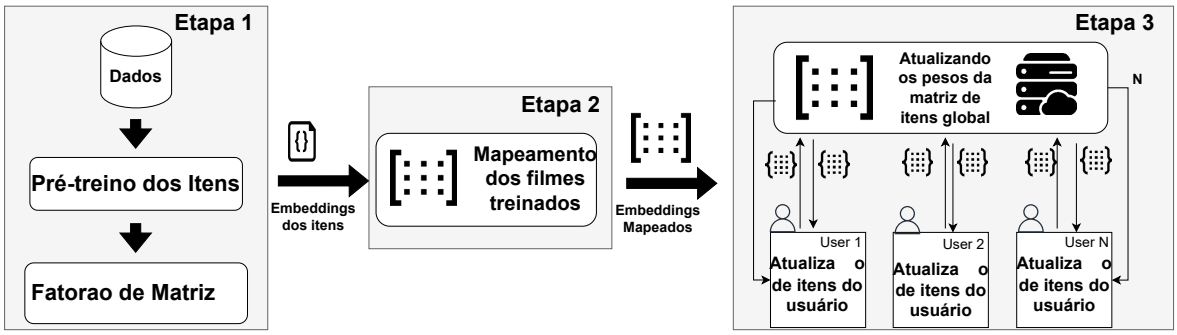

Figura 4.2: Estrutura do pipeline do experimento

Conforme ilustrado na Figura 4.2, nosso experimento é composto pelas seguintes etapas:

1. **Etapa 1:** Nesta etapa, realizamos o treinamento do sistema de recomendação centralizado. Além da inicialização randômica, utilizamos técnicas como PCA e Word2Vec, resultando em três pipelines distintos de treinamento centralizado. Para mais detalhes sobre como o processo foi conduzido, consulte a seção 4.3.

2. **Etapa 2:** Aqui, conduzimos o processo de mapeamento dos itens treinados na etapa anterior. Verificamos os filmes presentes em ambos os datasets e realizamos a transferência dos *embeddings*. Mais detalhes sobre este processo podem ser encontrados na seção 4.4.

3. **Etapa 3:** Esta é a etapa final do experimento, onde ocorre o treinamento do sistema de recomendação federado. Simulamos um ambiente real onde cada usuário mantém seus dados privados, compartilhando apenas a matriz de itens. Mais detalhes sobre este processo na seção 4.5.

Realizamos esse experimento para cada conjunto de dados utilizado. Isso significa que executamos um fluxo deste experimento para o Movielens 1B e repetimos todo o processo

para os dados da Netflix. Os conjuntos de dados constantemente utilizados na abordagem federativa foram o MovieLens 1M e o 100k.

4.3 Treinamento Centralizado

A fase de treinamento centralizado é o primeiro passo do nosso pipeline experimental, focada no desenvolvimento e validação de um modelo de recomendação por meio do treinamento em um conjunto de dados amplo. Durante esta etapa, aplicamos várias técnicas para aprimorar os *embeddings*, os quais são essenciais para a performance dos sistemas de recomendação. Os *embeddings* gerados nesta fase inicial são posteriormente empregados no treinamento do sistema de recomendação federado.

A Figura 4.3 ilustra o procedimento de treinamento do sistema de recomendação centralizado. O processo é dividido em três partes e começa com a leitura e validação dos dados. Na primeira etapa, os dados são lidos e submetidos a uma *5-fold cross validation*, onde o conjunto de dados é dividido em cinco partes iguais. Em cada rodada de validação, uma parte é utilizada como conjunto de teste e as quatro partes restantes como conjunto de treinamento. Isso é repetido cinco vezes, garantindo que cada parte dos dados seja usada tanto para treinamento quanto para teste. Esse procedimento gera conjuntos de treinamento e teste distribuídos aleatoriamente, com proporções de 80% e 20%, respectivamente.

A segunda etapa refere-se a à inicialização dos *embeddings*. Aqui, a técnica a ser utilizada para a atribuição inicial dos pesos aos *embeddings* pré-treinamento é selecionada. As alternativas são *PCA*, *Word2Vec* e Random. A inicialização varia conforme a técnica: para *Random*, os pesos são aleatórios, enquanto para *PCA* e *Word2Vec*, os *embeddings* são pré-treinados usando seus respectivos métodos.

Na terceira etapa, o foco é no treinamento do modelo de fatoração de matrizes. Utilizamos a implementação MF Cython como modelo principal, definindo $P, Q \sim \mathcal{N}(0, 0.1)$, $\alpha = 0.001$ (taxa de aprendizado) e $\lambda = 0.04$ (parâmetro de regularização) como valores padrão para os hiperparâmetros. Esses parâmetros foram escolhidos com base no estudo de Rendle et al. [48], que revisitou experimentos de filtragem colaborativa neural e fatoração de matrizes, demonstrando que uma configuração adequada desses hiperparâmetros pode melhorar substancialmente a qualidade do modelo. A taxa de aprendizado de 0.001 foi

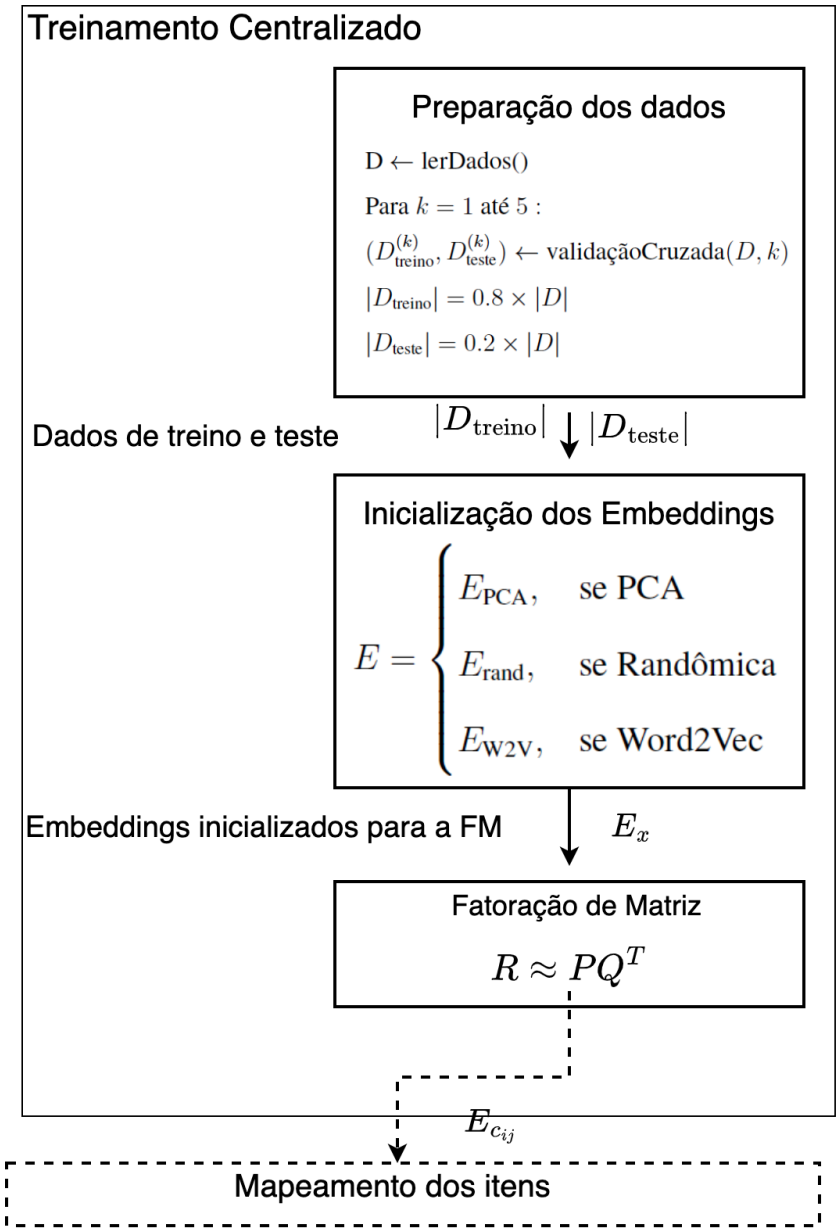

Figura 4.3: Estrutura do pipeline de Treinamento Centralizado

selecionada para garantir a convergência estável do modelo, enquanto o parâmetro de regularização de 0.04 foi escolhido para prevenir overfitting, garantindo que o modelo generalize bem para novos dados. Decidimos também desativar o viés do algoritmo para evitar que o viés do usuário influencie os *embeddings* gerados.

Após o treinamento, a matriz de *embeddings* dos itens é armazenada para uso posterior no mapeamento dos dados. Para cada conjunto de dados, é gerado um *embedding* utilizando cada uma das técnicas — *Random*, *PCA*, e *Word2Vec* —, possibilitando uma análise com-

parativa entre os *embeddings* resultantes dessas distintas abordagens. No caso do conjunto de dados MovieLens 1B, apenas a técnica *Word2Vec* é aplicada, devido ao *feedback* implícito presente nos dados que registram somente (i.e., sem rating) a interação usuário/item. O *Word2Vec* é apropriado para esta situação, uma vez que não considera a avaliação (*rating*) em seus *embeddings*, focando somente na interação.

4.4 Mapeamento dos Dados

Após estabelecer a base com o treinamento centralizado, uma etapa essencial que antecede a implementação efetiva da aprendizagem federada é o mapeamento dos *embeddings*. Essa etapa garante que os benefícios do treinamento centralizado sejam transferidos de forma eficaz e alinhada para o treinamento federado.

O processo de transferência de *embeddings* é crucial para garantir a eficiência do treinamento federado. Assim, nesta etapa, os *embeddings* treinados previamente são mapeados para os filmes presentes no dataset alvo: o treinado de forma centralizada e o destinado ao treinamento federado. Ressalta-se que todos os filmes utilizados no experimento possuem títulos em inglês.

Para o Movielens 1B, o mapeamento é direto, pois cada filme tem um ID único. Entretanto, a Netflix não usa os mesmos IDs que o Movielens. Portanto, adotamos uma estratégia de mapeamento tripartida:

1. Convertemos os títulos dos filmes para minúsculas e padronizamos a codificação de texto para UTF-8. Em seguida, tentamos uma correspondência exata usando o título e o ano de lançamento.

2. Se ainda houver filmes não mapeados, tentamos uma correspondência apenas pelo título.

3. Finalmente, replicamos o algoritmo de correspondência de strings proposto por Costa et al. [14] para emparelhar os filmes restantes. Apenas os matches com similaridade superior a 75% são aceitos, e somente o match com a maior similaridade é considerado para a transferência de *embeddings*.

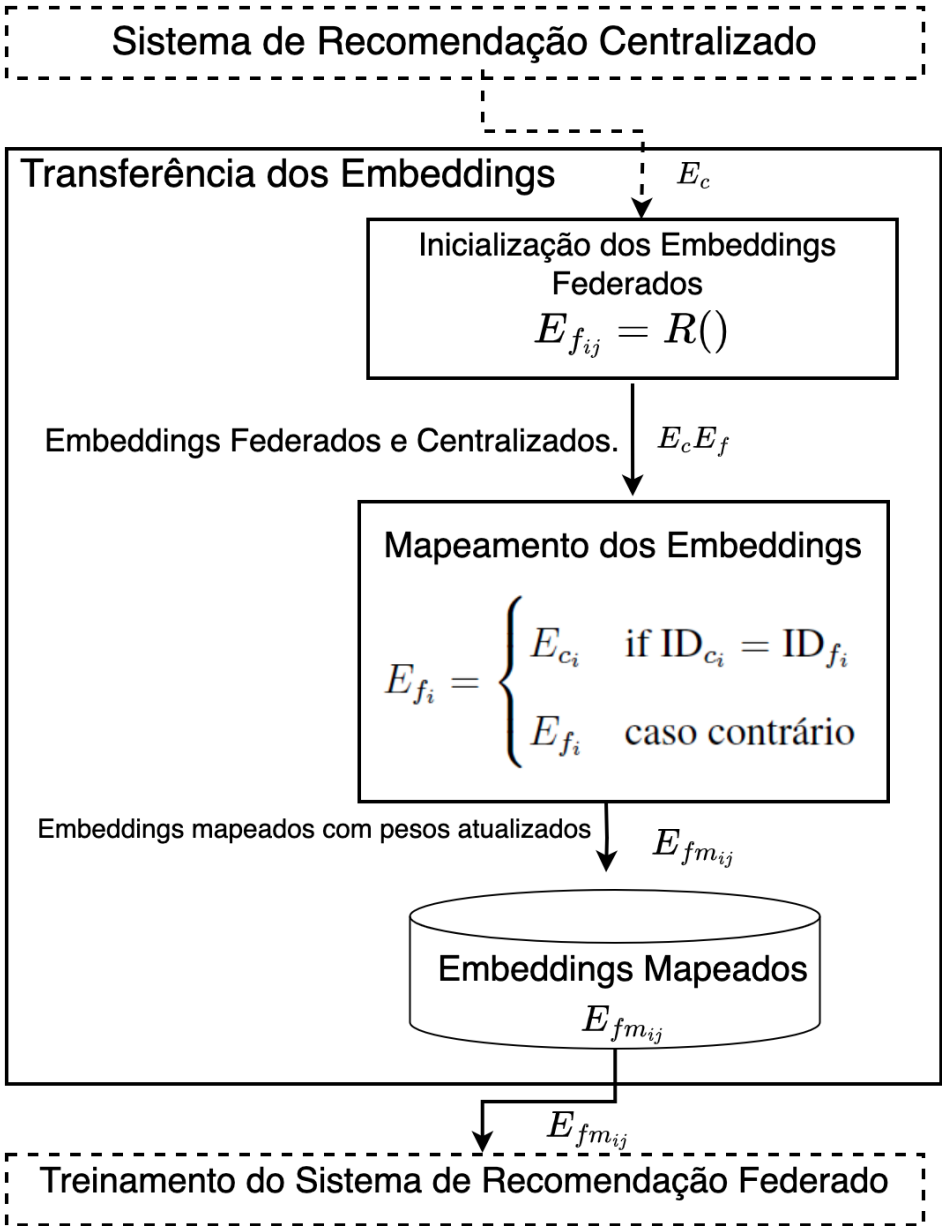

Figura 4.4: Estrutura do pipeline Transferência de *Embeddings*

A Figura 4.4 oferece uma visão detalhada deste processo de mapeamento. Posteriormente, os pesos dos *embeddings* são transferidos ou atualizados com base nas correspondências encontradas.

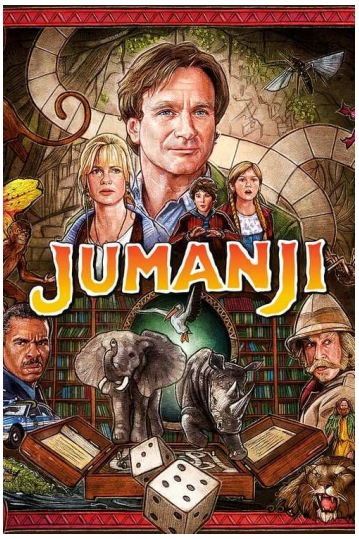

Figura 4.5: Jumanji 1995

Como exemplo, considere o filme *Jumanji*, lançado em 1995 (Figura 4.5). Ele está presente em ambos os datasets, o Movielens 1B e o Movielens 1M. Assim, é possível realizar a transferência de seus *embeddings*.

```
array([ 0.03169998,  0.03181237,  0.02390176,  0.03414932, -0.01296352,
        0.01502782,  0.01865523,  0.00047743, -0.00300238,  0.02073668,
       -0.0204319 , -0.03180487,  0.00343906,  0.00461743,  0.04066565,
       -0.00714946,  0.01060804, -0.00753643, -0.01823243, -0.02244614,
       -0.01480561, -0.0084219 , -0.00908006,  0.00487337, -0.00197657,
       -0.00080371,  0.01173264,  0.01856514, -0.00564357, -0.02613639,
       -0.02300347,  0.00051353, -0.04428071,  0.03441696,  0.01836052,
       -0.00669373,  0.00582858,  0.01264725,  0.03928597,  0.02950729,
        0.01134384,  0.00474346,  0.00391988, -0.00112446, -0.02065419,
       -0.04871101,  0.04782191,  0.04943485, -0.01341727,  0.04220046],
      dtype=float32)
```

Figura 4.6: Representação dos *embeddings* de Jumanji antes da transferência.

A Figura 4.6 mostra a distribuição inicial dos pesos de Jumanji, enquanto a Figura 4.7 ilustra os pesos após o processo de transferência.

```
array([ 0.03827822,  0.07743404, -0.17751467,  0.27137417, -0.17426483,
       -0.18524699,  0.29834706,  0.23875265, -0.22016022, -0.33786651,
       -0.05425733,  0.03923351, -0.5233169 , -0.3442587 , -0.37815002,
        0.01321101,  0.19115718, -0.16866808,  0.18776652, -0.11294346,
       -0.18765658,  0.2261571 , -0.30080095, -0.06786186, -0.10080723,
        0.04779506, -0.00544087, -0.31201607, -0.17979161,  0.19520359,
        0.15192248, -0.01951139, -0.34775776,  0.42747816,  0.49083817,
        0.42140204, -0.4752494 , -0.02046961,  0.22936815, -0.308858  ,
        0.04759337, -0.09237832,  0.05031604,  0.25508764,  0.15293059,
        0.15535033,  0.28771144, -0.4361507 ,  0.03525573,  0.18531272],
      dtype=float32)
```

Figura 4.7: Representação dos *embeddings* de Jumanji após a transferência.

Os filmes mapeados começam o treinamento federado com os pesos provenientes do processo de transferência, o que contribui significativamente para a eficiência e precisão do modelo federado final.

4.5 Treinamento Federado

Após mapear os dados conforme detalhado na seção anterior, passamos à terceira etapa da nossa metodologia, onde o foco reside no emprego da aprendizagem federada para o treinamento.

A importância desta técnica se torna ainda mais evidente ao considerarmos que, para sua aplicação prática necessitar de um aplicativo real e de múltiplos usuários, frameworks como TensorFlow Federated [5], PySyft [51], IBM FL [28], e FATE [4] possuem a capacidade de simular ambientes federados. Essas ferramentas oferecem uma base sólida para o desenvolvimento e teste de algoritmos em condições que se aproximam das encontradas em aplicações do mundo real.

Optamos pelo TensorFlow Federated, um framework do Google que se destacou como uma escolha sólida devido à sua maturidade, ampla documentação e as significativas contribuições de sua equipe no desenvolvimento da aprendizagem federada.

Quanto ao algoritmo de recomendação, optamos pelo *FMF FedRecon* desenvolvido por Karan Singhal et al. [57]. O *FMF FedRecon* destaca-se pela reconstrução de parâmetros locais para cada usuário individualmente, promovendo uma personalização precisa em sistemas de recomendação sem comprometer a privacidade dos dados. Esta abordagem confere ao *FMF FedRecon* eficácia e adaptabilidade. O procedimento operacional do *FMF FedRecon*

é estruturado nas etapas a seguir:

1. O servidor armazena e envia a matriz de itens I para os usuários selecionados em cada rodada;

2. Cada usuário "congela"I e treina o embedding do usuário U_u com um ou mais passos de SGD;

3. Os usuários, mantendo "congelado"U_u, treinam I através de SGD;

4. As atualizações de I são agregadas entre os usuários, preparando I para a rodada subsequente.

A natureza iterativa do *FMF FedRecon* significa que múltiplas rodadas de treinamento são realizadas até atingir a convergência desejada. Vale destacar que, em cada rodada, apenas uma amostra representativa de usuários é envolvida no treinamento. Isso captura cenários práticos, onde nem todos os dispositivos podem estar conectados simultaneamente.

O aumento no número de rodadas de comunicação entre os dispositivos e o servidor central implica em maior tempo de treinamento, maior consumo de banda de rede, e aumento dos custos operacionais. Esse cenário pode resultar em atrasos na convergência do modelo e em um desgaste da experiência do usuário, especialmente em ambientes onde a largura de banda é limitada.

A otimização neste contexto refere-se à minimização do número de rodadas de comunicação necessárias para alcançar uma convergência eficiente do modelo, bem como à melhoria na precisão das previsões por meio de técnicas avançadas de transferência de aprendizado.

4.6 Arquitetura Federada

Após discutirmos os detalhes do processo de treinamento federado na seção anterior, é fundamental abordarmos a estrutura subjacente que sustenta tais treinamentos: a arquitetura federada.

Hongyi Zhang et al. [64] exploraram quatro arquiteturas distintas para sistemas federados com o principal objetivo de testar a replicabilidade dessas arquiteturas em cenários de

produção. Isso visa auxiliar os desenvolvedores na seleção da estrutura que melhor atende às necessidades de seu projeto.

Eles se concentraram nas seguintes arquiteturas: centralizada, hierárquica, regional e descentralizada. Através de um experimento com um modelo de detecção de imagens, os autores simularam ambientes com diferentes números de usuários. Em cada simulação, uma arquitetura foi posta à prova, e métricas como latência de rede, acurácia dos modelos e tempo de treinamento foram analisadas em cada iteração. Essa investigação permitiu que os autores oferecessem uma comparação concisa entre as diferentes arquiteturas.

Com base em nossa análise das arquiteturas, optamos pela centralizada. A razão desta escolha se fundamenta na necessidade de manter o controle da matriz de itens e do processo de agregação para todos os usuários. Isso é crucial já que todos os usuários precisam ter acesso a todos os itens no sistema de recomendação.

Capítulo 5

Resultados

Este capítulo avalia a eficácia da abordagem proposta para aprimorar o aprendizado federado em sistemas de recomendação, com ênfase na redução do custo de comunicação e na melhoria da eficácia da convergência. Inicialmente, investigamos como a transferência de embeddings pode diminuir o número de rodadas necessárias para alcançar a convergência no *FMF FedRecon* , utilizando conjuntos de dados como MovieLens 1B e Netflix. A análise foca no impacto dessa transferência nos processos de convergência em ambientes federados, proporcionando uma avaliação detalhada dos benefícios e limitações da nossa abordagem.

Segue-se a discussão sobre a redução na quantidade de usuários por rodada de treinamento, visando manter o desempenho do sistema com menor interação. Este aspecto é importante para implementações práticas em ambientes reais, onde a eficiência operacional é muitas vezes limitada por recursos como largura de banda e tempo de processamento. Em cenários do mundo real, a minimização da necessidade de comunicação direta entre o servidor e os dispositivos dos usuários pode significativamente melhorar a eficiência do sistema. Além disso, dadas as restrições práticas como disponibilidade de dispositivo e conectividade, uma abordagem que exige menos usuários por rodada facilita um modelo federado mais escalável e viável para adoção em larga escala.

O estudo também examina como a seleção estratégica de embeddings afeta a rapidez da convergência federada, fornecendo insights sobre o papel da transferência de dados na otimização do aprendizado federado.

Avaliamos a eficiência dos sistemas de recomendação em conformidade com normas de privacidade, como a LGPD, mostrando que a transferência de embeddings pode manter a

qualidade das recomendações sem violar a privacidade dos dados dos usuários. Este capítulo destaca abordagens metodológicas para enfrentar desafios no aprendizado federado e demonstra sua aplicabilidade em melhorar a comunicação, acelerar a convergência do modelo e aderir à legislação de privacidade.

Por último, os experimentos foram conduzidos em um ambiente simulado utilizando o Google Colab, com o framework *TensorFlow Federated* configurado para replicar um ambiente federado. Esta configuração permitiu testar o comportamento do sistema em condições próximas às reais.

5.1 Redução de Comunicação

O objetivo deste experimento é avaliar se conseguimos reduzir o custo de comunicação do *FMF FedRecon* , ao avaliar o impacto da transferência de *embeddings* na redução do número de rodadas requeridas para alcançar a convergência. O critério de comparação adotado é o valor de *RMSE* = 0.907, conforme reportado no estudo original do *FMF FedRecon*, com a preservação do mesmo número de usuários por rodada, ou seja, 50. Para tal, utilizamos o conjunto de dados *MovieLens 1M*, o mesmo empregado na pesquisa do *FMF FedRecon*, com a dimensionalidade dos *embeddings* fixada em 50. Nesse contexto, o custo de comunicação refere-se ao total de rodadas necessárias para treinar o modelo, indicando que uma menor quantidade de rodadas, e consequentemente, uma menor necessidade de interação com os usuários, representa um cenário mais eficiente para o treinamento.

5.1.1 Sistemas de Recomendação centralizados com dados reais

Contextualização do Experimento

Essa seção apresenta o experimento associado à Questão de Pesquisa 1 (QP1) de nossa pesquisa. Esta pergunta tem como principal objetivo investigar o impacto da transferência de *embeddings* provenientes do treinamento centralizado no conjunto de dados da Netflix para otimizar processos em contextos federados.

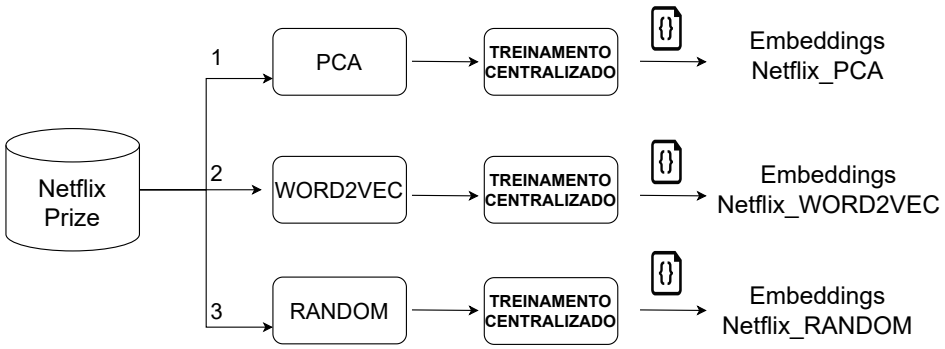

Figura 5.1: Fluxo do experimento com os dados da *Netflix*

Treinamento Centralizado e Objetivo

Inicialmente, conduzimos um treinamento centralizado empregando os dados originários da Netflix. Conforme ilustrado na Figura 5.1, o processo foi executado em três iterações distintas. A meta subjacente a essa abordagem foi a produção de diferentes *embeddings*, todos derivados dos mesmos dados, porém variando as técnicas de inicialização. Tal variação visa aferir os potenciais impactos na aprendizagem federativa.

Mapeamento dos IDs dos Filmes

Uma vez concluído o treinamento centralizado, procedemos ao mapeamento dos IDs dos filmes, procedimento este detalhado na Seção 4.4. Durante essa fase, foi alcançada uma taxa de mapeamento de aproximadamente 73%. Isso significa que, do total de 3.706 filmes catalogados no conjunto de dados do MovieLens 1M, identificamos 2.700 filmes que também estão presentes na base da Netflix.

Processo de Transferência

Com os IDs devidamente mapeados, avançamos para a fase de transferência, que envolve a identificação de filmes por meio dos IDs comuns entre os conjuntos de dados, permitindo assim a transferência dos pesos dos *embeddings*. Para aqueles filmes que não foram localizados nos dados da Netflix, procedemos com a inicialização de seus pesos de forma aleatória, adotando uma técnica específica de normalização. Essa técnica implica a inicialização dos pesos dos *embeddings* através de valores gerados aleatoriamente, seguindo uma distribuição normal com média zero.

Treinamento Federado

A fase subsequente envolveu o treinamento federado. Durante esta etapa, mantivemos o protocolo padrão do treinamento base do *FedRecon*. Em cada rodada de treinamento, uma amostra aleatória composta por 50 usuários foi selecionada para aprimorar o modelo.

Para garantir a comparabilidade direta com o estudo original, utilizamos o mesmo conjunto de dados de avaliação, o MovieLens 1M. A divisão dos dados foi feita da mesma forma que no estudo original: 80% para treinamento, 10% para validação e 10% para teste, utilizando timestamps para criar essas divisões. Adotamos como métrica de referência o valor de RMSE = 0.907, conforme exposto no estudo original.

Resultado e Discussão

Metodo	**Rodadas**	**Clients**	**RMSE**
FMF FedRecon	500	50	≈ 0.907
Netflix_Random	30	50	≈ 0.906
Netflix_PCA	28	50	≈ 0.903
Netflix_Word2Vec	27	50	≈ 0.904

Tabela 5.1: Resultado do Experimento com os dados da *Netflix*

Na análise realizada dos resultados evidenciados na Tabela 5.2, constatou-se que a técnica de transferência de *embeddings* de itens tem uma boa capacidade de otimizar a convergência do modelo. Especificamente, esta abordagem exibiu uma diminuição expressiva no número de rodadas requeridas para atingir a convergência. Ao comparar-se com o método *Netflix_Random*, que, mesmo empregando a transferência, necessitou de um número superior de rodadas, observou-se uma redução significativa de 94%.

Adicionalmente, ao avaliar a performance de diferentes técnicas de inicialização dos *embeddings*, identificou-se uma variação sutil tanto no número de rodadas quanto no RMSE. A técnica de inicialização com *PCA* apresentou o menor RMSE, seguida pela técnica *Word2Vec* e, por fim, pela técnica tradicional *Random*. Além disso, houve uma redução no número de rodadas de 30 para 27 entre a técnica Random e Word2Vec. Essa observação sugere que a

escolha da técnica de inicialização dos *embeddings* pode influenciar a eficiência e a precisão do modelo, e merece uma análise mais detalhada em pesquisas futuras.

5.1.2 Sistemas de Recomendação centralizados com dados sinteticos

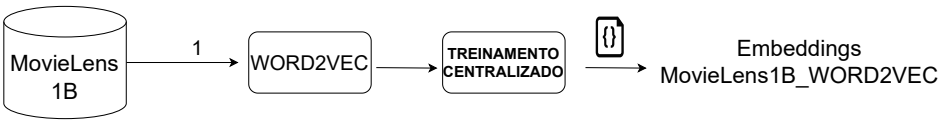

Figura 5.2: Fluxo do experimento com os dados *MovieLens* 1B

Contextualização do Experimento

Esta seção discorre sobre o experimento associado à Questão de Pesquisa 2 (QP2) de nossa pesquisa. Aqui, queremos investigar o impacto da transferência de *embeddings* provenientes do treinamento centralizado no conjunto de dados sintéticos do MovieLens 1B para otimizar processos em contextos federados.

Treinamento Centralizado e Objetivo

Inicialmente, conduzimos um treinamento centralizado empregando os dados originários do MovieLens 1B. Conforme ilustrado na Figura 5.2, o processo foi executado em um único fluxo. Isso ocorre porque os dados do MovieLens 1B são de feedback implícito; ou seja, possuímos apenas informações sintéticas de se um usuário assistiu a um determinado filme. Como explicado na seção 4.3, a técnica de Word2Vec independe do rating que utilizamos nas demais técnicas para inicializar os *embeddings*. Desse modo, utilizamos esta para treinar o modelo e gerar os *embeddings* para o treinamento federativo.

Mapeamento dos IDs dos Filmes

Uma vez concluído o treinamento centralizado, não precisamos mapear os IDs dos filmes. Como se trata de um conjunto de dados que possui o mesmo ID para os filmes, essa etapa não se faz necessária. No entanto, tratando-se de identificação, 98% dos filmes do conjunto de dados alvo, ou seja, do MovieLens 1M, foram encontrados no MovieLens 1B.

Processo de Transferência

Com os identificadores (IDs) dos dados corretamente mapeados, deu-se início à fase de transferência. Tal fase se caracteriza pela identificação dos filmes utilizando os IDs que se encontram em ambos os conjuntos de dados, resultando na subsequente transferência dos pesos associados aos *embeddings*. Filmes não identificados no conjunto de dados MovieLens 1B tiveram seus pesos inicializados de maneira aleatória, seguindo uma função de normalização previamente definida. Como consequência desse processo, um total de 69 filmes conservou os pesos de seus *embeddings* conforme estabelecido pela metodologia original.

Treinamento Federado

A etapa subsequente dedicou-se ao treinamento federado. Durante esse processo, aderimos ao protocolo padrão do treinamento baseado no FMF FedRecon como demostrado no experimento anterior.

Resultado e Discursão

Metodo	**Rodadas**	**Clients**	**RMSE**
FMF FedRecon	500	50	≈ 0.907
MovieLens1B_Word2vec	2	50	≈ 0.888

Tabela 5.2: Resultado do Experimento com os dados sinteticos

Conforme apresentado na Tabela 5.2, é significativa diminuição no número de rodadas requeridas para se alcançar a convergência, evidenciando uma redução superior a 99%. Ao analisarmos a performance de nossa proposta, observa-se um RMSE aproximado de 0.888.

É importante destacar que os dados empregados para a geração dos *embeddings* foram produzidos de maneira artificial. Tal aspecto ratifica que, em nossa proposta metodológica, é viável integrar esse processo de geração de dados sintéticos ao pipeline.

Assim, sistemas que enfrentam restrições de dados, como aqueles em que os dados disponíveis são escassos, dispersos ou difíceis de coletar, podem adotar essa estratégia de geração de dados sintéticos para compensar essas limitações. Isso não só facilita a criação de uma base essencial para o treinamento federado, mas também potencializa a capacidade do

sistema de alcançar uma performance comparável à obtida com dados reais, reduzindo significativamente o número de rodadas necessárias para a convergência e, consequentemente, otimizando o uso de recursos computacionais. Além disso, ao integrar a geração de embeddings a partir de dados sintéticos no pipeline, a abordagem proposta demonstra sua eficácia e adaptabilidade em cenários onde a obtenção de dados reais pode ser um desafio.

5.2 Redução de Número de Usuários

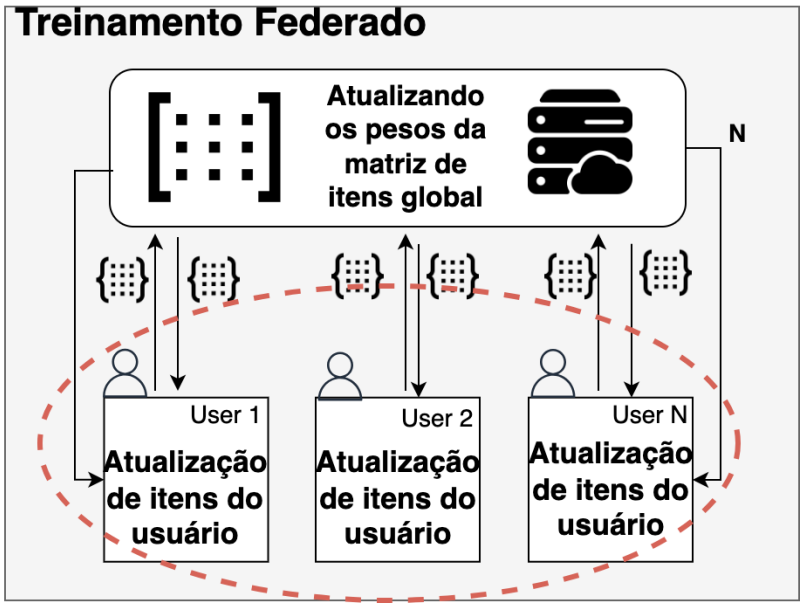

Figura 5.3: Pipiline do Experimento de Redução de Usuários

5.2.1 Objetivo do Experimento

O experimento em questão está relacionado à Questão de Pesquisa 3 (QP3) proposta neste trabalho. Seu principal objetivo é investigar a possibilidade de, mesmo com uma quantidade reduzida de usuários, manter uma diminuição efetiva da comunicação. Em termos práticos, busca-se alcançar um número de rodadas necessário para a convergência que seja inferior a 500 ou seja inferior ao número de rodadades necessários para o original atingir sua convergência como demostrado no artigo [57], contudo, empregando uma menor quantidade de usuários em cada rodada.

5.2.2 Contextualização e Metodologia

Ao discutirmos a redução de usuários, estamos nos referindo especificamente à quantidade de usuários necessários por rodada para executar o treinamento federado, conforme ilustrado na Figura 5.3. Nos experimentos anteriores, conforme explicado no artigo original do FMF FedRecon , foi demonstrado o uso de 50 usuários, selecionados aleatoriamente por rodada, para a execução do treinamento. Agora, a proposta atual busca investigar a possibilidade de reduzir esse número sem comprometer o processo de convergência, utilizando a transferência dos *embeddings* dos itens como estratégia para alcançar esse objetivo.

5.2.3 Implementação e Dados Utilizados

Para a concretização deste experimento, recorremos aos *embeddings* gerados na etapa 5.1 deste estudo. Os *embeddings* em questão incluem: Netflix_Random, Netflix_PCA, Netflix_Word2Vec e MovieLens1B_Word2Vec. O diferencial metodológico neste experimento reside na execução de múltiplos treinamentos de modelos distintos, todos fazendo uso da transferência dos *embeddings*. Esta abordagem envolve decrementar de 10 em 10 o número de usuários envolvidos na amostragem. Desse modo, foram realizados treinamentos federativos com 40, 30, 20 e 10 usuários amostrados. Para cada cenário estabelecido, foram conduzidos quatro treinamentos diferentes, considerando os *embeddings* previamente mencionados. Todas as instâncias de treinamento, empregou-se uma validação cruzada com k igual a 5.

5.2.4 Resultados

As tabelas abaixo apresentam os resultados do experimento com diferentes amostragens de usuários, a saber: 40, 30, 20 e 10 usuários.

Metodo	**Rodadas**	**Clients**	**RMSE**
Netflix_Random	31	40	≈ 0.903
Netflix_PCA	27	40	≈ 0.905
Netflix_Word2Vec	27	40	≈ 0.903
MovieLens1B_Word2Vec	2	40	≈ 0.891

Tabela 5.3: Tabela comparativa para a amostra de 40 usuários

Na Tabela 5.3, constata-se que, mesmo reduzindo a quantidade de usuários, foi possível manter uma comunicação inferior a 500 rodadas para a convergência. A diminuição no número de usuários resultou em um aumento discreto no número de rodadas necessárias, entretanto, o processo ainda permaneceu dentro do objetivo estabelecido de redução.

Metodo	**Rodadas**	**Clients**	**RMSE**
Netflix_Random	32	30	≈ 0.904
Netflix_PCA	30	30	≈ 0.904
Netflix_Word2Vec	30	30	≈ 0.902
MovieLens1B_Word2Vec	2	30	≈ 0.893

Tabela 5.4: Tabela comparativa para a amostra de 30 usuários

A Tabela 5.4 ilustra uma redução de 10 usuários em relação ao experimento anterior, ainda assim, mantendo a comunicação abaixo das 500 rodadas para convergência.

Metodo	**Rodadas**	**Clients**	**RMSE**
Netflix_Random	35	20	≈ 0.906
Netflix_PCA	33	20	≈ 0.903
Netflix_Word2Vec	33	20	≈ 0.904
MovieLens1B_Word2Vec	3	20	≈ 0.896

Tabela 5.5: Tabela comparativa para a amostra de 20 usuários

Em comparação com os resultados apresentados na Tabela 5.4, a Tabela 5.5 evidencia uma consistência nos métodos, mesmo com uma amostra 30 usuários menor. Nota-se que

alguns métodos mantiveram-se constantes no número de rodadas necessárias para a convergência.

Metodo	**Rodadas**	**Clients**	**RMSE**
Netflix_Random	45	10	≈ 0.902
Netflix_PCA	40	10	≈ 0.905
Netflix_Word2Vec	37	10	≈ 0.906
MovieLens1B_Word2Vec	5	10	≈ 0.897

Tabela 5.6: Tabela comparativa para a amostra de 10 usuários

Por fim, a Tabela 5.6 demonstra que, mesmo com uma amostra drasticamente reduzida para 10 usuários, os resultados permaneceram satisfatórios, com uma convergência abaixo de 0,907. Em comparação com a abordagem original, sem a implementação do processo de transferência, o número de rodadas foi reduzido para menos de 10% do inicialmente necessário. Especificamente, no caso dos *embeddings* gerados pelo MovieLens 1B, houve uma redução significativa de 80% no número de usuários e 99% no número de rodadas necessárias para atingir a convergência, em relação ao método original FMF FedRecon.

A principal diferença entre os resultados do Netflix e do MovieLens 1B se deve ao processo de matching. O MovieLens 1B apresenta um maior número de matchings, ou seja, mais filmes tiveram seus *embeddings* inicializados com os pesos treinados na etapa centralizada, ao contrário da Netflix, onde o número de correspondências foi menor. Essa diferença na quantidade de filmes que puderam aproveitar os *embeddings* previamente treinados contribui para os resultados observados no MovieLens 1B. Esses resultados evidenciam o ganho obtido com a abordagem de transferência de *embeddings*, mostrando que a maior eficiência no matching impacta diretamente a quantidade de rodadas nescessarias de treinamento.

A redução no número de clientes necessários permite uma maior escalabilidade do sistema, eliminando a dependência de um grande volume de clientes conectados para o treinamento.

5.3 Impacto da Amostragem de *Embeddings* na Convergência Federada

Nesta seção, abordamos o experimento relacionado à Questão de Pesquisa 4 (QP4). O principal objetivo deste experimento é investigar o impacto da amostragem de itens no treinamento federativo. Mais especificamente, buscamos entender as diferenças entre transferir todos os itens versus transferir apenas uma amostra deles e, consequentemente, determinar após quantas rodadas ocorre a convergência.

Nossa intenção é avaliar qual é o cenário mais propício: se é mais vantajoso possuir uma amostra dos itens alvo para treinamento centralizado ou, conforme discutido anteriormente, se é preferível realizar uma amostragem dos itens.

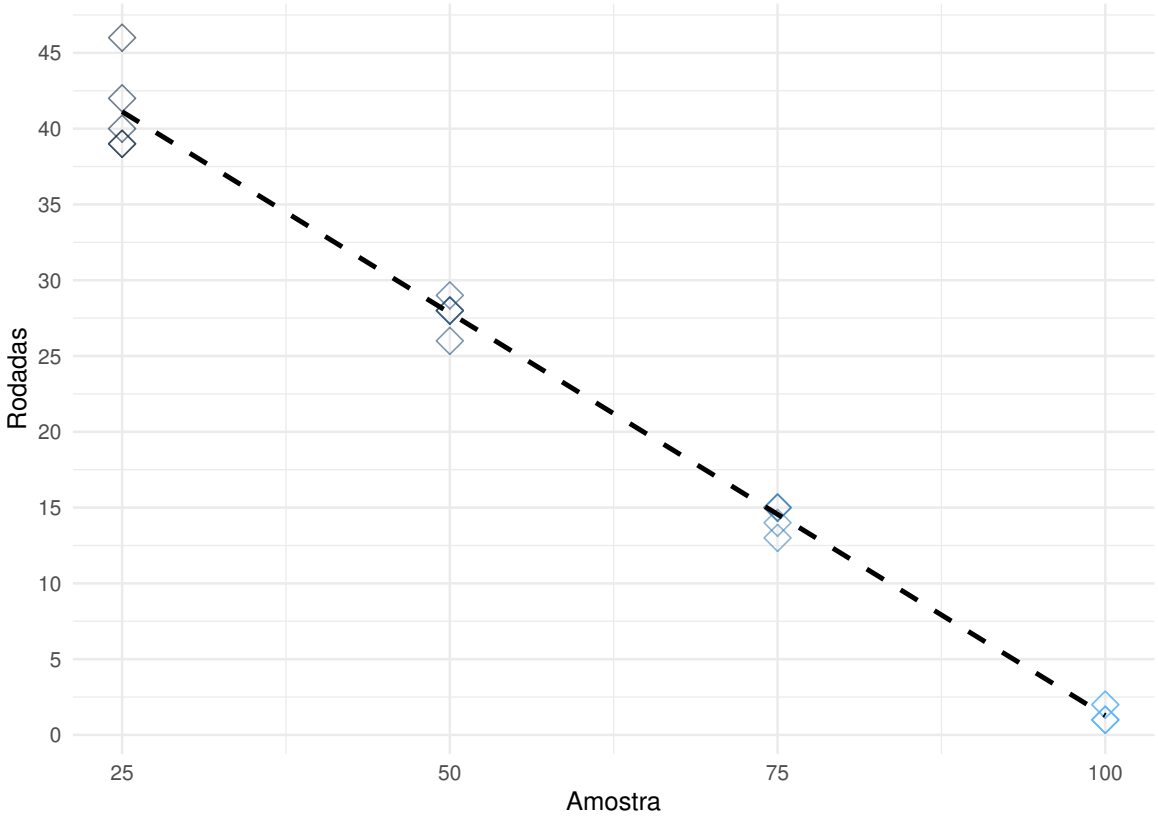

Figura 5.4: Resultados comparativos entre rodadas usando amostra de itens.

Para concretizar este experimento, adotamos uma abordagem de validação cruzada com k=5. Mantivemos um número constante de 50 usuários, selecionados aleatoriamente por rodada de treinamento, e definimos um valor de referência RMSE de 0.907. Implementamos

quatro cenários distintos; em cada um, uma determinada porcentagem de itens foi selecionada, correspondendo a uma fração dos itens previamente treinados de forma centralizada. Posteriormente, analisamos em quantas rodadas a convergência era alcançada. Em todas as variações, empregamos os *embeddings* do conjunto de dados Movielens 1B, que, como evidenciado em experimentos anteriores, produziu os melhores resultados.

5.3.1 Resultados

A análise do Gráfico 5.4 sugere que a porcentagem de itens transferidos para o treinamento influencia diretamente o número de rodadas necessárias para a convergência. Por exemplo, com apenas 25% dos itens, aproximadamente 40 rodadas são necessárias para atingir a convergência. Ao avaliar até 75% dos itens, ainda percebemos uma quantidade significativa de rodadas, mas com uma performance superior à obtida com os dados da Netflix ou sem qualquer processo de transferência. Curiosamente, ao transferirmos somente 50% dos itens, a média de rodadas se assemelha ao resultado obtido com os dados da Netflix, evidenciando a eficácia da amostragem sintética neste contexto.

5.4 Eficiência do Sistema de Recomendação sob Conformidade de Privacidade

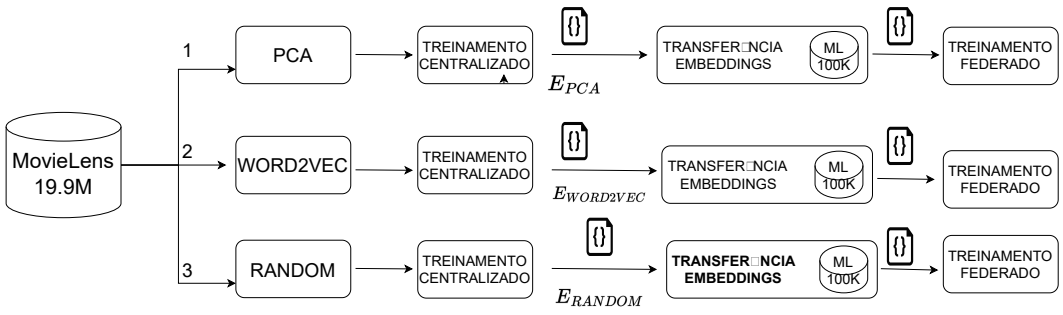

Figura 5.5: Pipiline do Experimento Sustentando Desempenho em Recomendações

5.4.1 Objetivo do Estudo de caso

O principal objetivo deste experimento é entender como uma plataforma pode manter seu desempenho e gerar recomendações para um usuário, mesmo preservando a privacidade e integridade de seus dados no contexto da LGPD.

5.4.2 Contextualização

Com a crescente preocupação em relação à privacidade dos dados dos usuários, empresas têm sido pressionadas a adaptar suas abordagens de coleta e processamento de informações. A promulgação da LGPD elevou ainda mais estas preocupações, concedendo aos usuários o direito de não compartilhar ou até mesmo de solicitar a remoção completa de seus registros dos sistemas das empresas. Dada esta nova conjuntura, torna-se imperativo repensar os métodos pelos quais os sistemas de recomendação são treinados e implementados, garantindo tanto a eficácia quanto a conformidade com as regulamentações de privacidade.

5.4.3 Metodologia, Implementação e Dados Utilizados

Para explorar esta problemática, foi conduzido um experimento como mostrado na Figura 5.5 empregando o conjunto de dados amplamente reconhecido: o MovieLens 20M. A partir deste, uma subamostra equivalente ao MovieLens 100K foi extraída, simbolizando o segmento de usuários que deseja remover suas informações do servidor centralizado e mantê-las exclusivamente em dispositivos pessoais.

No ambiente de treinamento centralizado, o conjunto de dados do MovieLens 20M (excluindo a subamostra 100K) foi utilizado. Em contraste, no cenário federativo, recorreu-se ao conjunto de dados correspondente ao MovieLens 100K, representando os usuários que optaram por uma maior restrição de dados.

A abordagem federativa se distinguiu pela incorporação de uma matriz de item mapeada com *embeddings* atualizados, visando avaliar a capacidade de manter a performance em relação ao modelo centralizado tradicional.

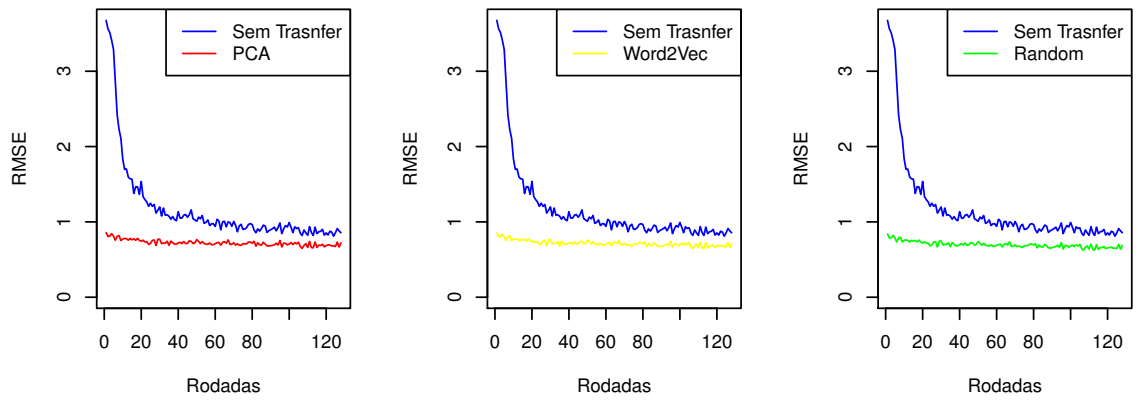

Figura 5.6: Resultados comparativos entre rodadas usando o conjunto de dados MovieLens 100K

5.4.4 Resultados

A análise dos resultados, esquematizada na Figura 5.6, oferece insights reveladores. Ao confrontar estratégias que incorporam a transferência de *embeddings* com aquelas que prescindem deste recurso, identifica-se uma vantagem notável na eficiência da primeira. Este achado sugere que a proposta deste trabalho opera com distinção, mantendo a integridade do sistema de recomendação preestabelecido, mesmo em cenários onde o usuário opta por uma postura mais restritiva quanto à partilha de seus dados. Note que os *embeddings* transferidos são oriundos exclusivamente dos registros dos usuários que consensualmente compartilham suas informações de forma centralizada. Desde os estágios iniciais, percebe-se uma superioridade na performance desta abordagem em relação ao treinamento do sistema a partir de uma base nula. Adicionalmente, após 120 rodadas, a metodologia desprovida de transferência não evidencia sinais de convergência para um RMSE <= 0.94, consolidando a robustez e pertinência da abordagem proposta nesta dissertação.

Capítulo 6

Conclusões e Trabalhos Futuros

Este trabalho explorou métodos para aprimorar o aprendizado federado em sistemas de recomendação, com foco na redução do custo de comunicação e na eficácia do processo de convergência. Através da transferência de embeddings e da redução do número de usuários por rodada, identificamos estratégias que podem melhorar significativamente a eficiência e a viabilidade dos sistemas federados. A investigação demonstrou o potencial da transferência de embeddings para acelerar a convergência e otimizar a comunicação em ambientes federados, oferecendo uma alternativa promissora às abordagens convencionais.

Além disso, este estudo ressaltou a importância da conformidade com as normas de privacidade, como a LGPD, mostrando que é possível manter a eficácia das recomendações sem comprometer a privacidade dos usuários. A utilização do *FedRecon*, em particular, foi fundamental para explorar a reconstrução de parâmetros locais, enfatizando a personalização e a privacidade nos sistemas de recomendação federados.

Para trabalhos futuros, é fundamental explorar modelos alternativos ao *FedRecon*, ampliando o espectro de técnicas federadas para otimização de sistemas de recomendação. Modelos como o *FedAvg with Differential Privacy* e o *FedProx*, ambos discutidos no Capítulo 3, representam direções promissoras. O *FedAvg with Differential Privacy* oferece melhorias significativas na proteção da privacidade do usuário, tornando o sistema mais seguro em ambientes sensíveis a dados pessoais, enquanto o *FedProx* aborda problemas de heterogeneidade de dados entre os dispositivos, melhorando a convergência e a estabilidade do treinamento federado. Explorar essas e outras abordagens pode não apenas aumentar a eficiência e precisão das recomendações, mas também promover a adoção do aprendizado federado em

uma gama mais ampla de contextos e aplicações. Investigar a aplicabilidade dessas estratégias em diferentes domínios e configurações de dados reforçará a robustez e flexibilidade dos sistemas de recomendação federados, abrindo caminho para avanços significativos na área.

Bibliografia

[1] Z. Alamgir, F.K. Khan, and S. Karim. Federated recommenders: methods, challenges and future. *Cluster Computing*, 25:4075–4096, 2022.

[2] Waqar Ali, Rajesh Kumar, Zhiyi Deng, Yansong Wang, and Jie Shao. A Federated Learning Approach for Privacy Protection in Context-Aware Recommender Systems. *The Computer Journal*, 64(7):1016–1027, 04 2021.

[3] Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Antonio Ferrara, and Fedelucio Narducci. Federank: User controlled feedback with federated recommender systems, 2021.

[4] T. F. Authors. Federated ai technology enabler, 2019.

[5] T. T. Authors. Tensorflow federated, 2019.

[6] Oren Barkan and Noam Koenigstein. Item2vec: Neural item embedding for collaborative filtering, 2017.

[7] Francois Belletti, Karthik Lakshmanan, Walid Krichene, Yi-Fan Chen, and John Anderson. Scalable realistic recommendation datasets through fractal expansions, 2019.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[9] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics*, pages 177–186, 2010.

[10] Nader Bouacida and Prasant Mohapatra. Vulnerabilities in federated learning. *IEEE Access*, 9:63229–63249, 2021.

[11] Zeyu Cao, Zhipeng Liang, Bingzhe Wu, Shu Zhang, Hangyu Li, Ouyang Wen, Yu Rong, and Peilin Zhao. Privacy matters: Vertical federated linear contextual bandits for privacy protected recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 154–166, New York, NY, USA, 2023. Association for Computing Machinery.

[12] Di Chai, Leye Wang, Kai Chen, and Qiang Yang. Secure federated matrix factorization. *IEEE Intelligent Systems*, 36(5):11–20, September 2021.

[13] Chaochao Chen, Liang Li, Bingzhe Wu, Cheng Hong, Li Wang, and Jun Zhou. Secure social recommendation based on secret sharing. *ArXiv*, abs/2002.02088, 2020.

[14] Júlio B. G. Costa, Leandro B. Marinho, Rodrygo L. T. Santos, and Denis Parra. Evaluating pre-training strategies for collaborative filtering. In *Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '23, page 175–182, New York, NY, USA, 2023. Association for Computing Machinery.

[15] Jie Ding, Eric Tramel, Anit Kumar Sahu, Shuang Wu, Salman Avestimehr, and Tao Zhang. Federated learning challenges and opportunities: An outlook. In *ICASSP 2022*, 2022.

[16] John Doe and Jane Smith. The efficacy of recommendation systems: A comparative study. *Journal of Modern Tech and Computing*, 34(2):34–56, March 2022.

[17] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[18] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

[19] Chen Gao, Chao Huang, Dongsheng Lin, Depeng Jin, and Yong Li. Dplcf: Differentially private local collaborative filtering. In *Proceedings of the 43rd International ACM*

SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, page 961–970, New York, NY, USA, 2020. Association for Computing Machinery.

[20] Yoav Goldberg. Neural network methods for natural language processing. 2017.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

[23] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. Social media recommendation based on people and tags. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 194–201. ACM, 2010.

[24] P. Hanafizadeh, M. Barkhordari Firouzabadi, and K.M. Vu. Insight monetization intermediary platform using recommender systems. *Electron Markets*, 31:269–293, 2021.

[25] F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), dec 2015.

[26] István Hegedüs, Gábor Danner, and Márk Jelasity. Decentralized recommendation based on matrix factorization: A comparison of gossip and federated learning. In *PKDD/ECML Workshops*, 2019.

[27] Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 2nd edition, 2006.

[28] IBM. Ibm federated learning, 2020.

[29] Junjie Jia and Zhipeng Lei. Personalized recommendation algorithm for mobile based on federated matrix factorization. *Journal of Physics: Conference Series*, 1802:032021, 03 2021.

[30] Farwa K. Khan, Adrian Flanagan, Kuan Eeik Tan, Zareen Alamgir, and Muhammad Ammad-ud din. A payload optimization method for federated recommender systems.

In *Proceedings of the 15th ACM Conference on Recommender Systems*, RecSys '21, page 432–442, New York, NY, USA, 2021. Association for Computing Machinery.

[31] Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 426–434, New York, NY, USA, 2008. Association for Computing Machinery.

[32] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[33] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[34] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks, 2009.

[35] Tan Li, Linqi Song, and Christina Fragouli. Federated recommendation system via differential privacy. In *2020 IEEE International Symposium on Information Theory (ISIT)*, pages 2592–2597, 2020.

[36] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, may 2020.

[37] Zhen Liao, Yang Song, Li-wei He, and Yalou Huang. Evaluating the effectiveness of search task trails. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 489–498, New York, NY, USA, 2012. Association for Computing Machinery.

[38] Yujie Lin, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Dongxiao Yu, Jun Ma, Maarten de Rijke, and Xiuzhen Cheng. Meta matrix factorization for federated rating predictions. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 981–990, New York, NY, USA, 2020. Association for Computing Machinery.

[39] Weiming Liu, Chaochao Chen, Xinting Liao, Mengling Hu, Jianwei Yin, Yanchao Tan, and Longfei Zheng. Federated probabilistic preference distribution modelling with compactness co-clustering for privacy-preserving multi-domain recommendation. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 2206–2214. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

[40] Leandro Balby Marinho, Júlio Barreto Guedes da Costa, Denis Parra, and Rodrygo L. T. Santos. Similarity-based explanations meet matrix factorization via structure-preserving embeddings. In *27th International Conference on Intelligent User Interfaces*, IUI '22, page 782–793, New York, NY, USA, 2022. Association for Computing Machinery.

[41] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2017.

[42] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.

[43] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.

[44] Khalil Muhammad, Qinqin Wang, Diarmuid O'Reilly-Morgan, Elias Tragos, Barry Smyth, Neil Hurley, James Geraci, and Aonghus Lawlor. Fedfast: Going beyond average for faster training of federated recommender systems. page 1234–1242, 2020.

[45] Andrew Y Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[46] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pretraining. 2018.

[48] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. Neural collaborative filtering vs. matrix factorization revisited. In *RecSys '20: Fourteenth ACM Conference on Recommender Systems*, Virtual Event, Brazil, September 2020. ACM.

[49] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, page 175–186, New York, NY, USA, 1994. Association for Computing Machinery.

[50] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[51] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning, 2018.

[52] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 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. ACM, 2002.

[53] Bichen Shi, Elias Z. Tragos, Makbule Gulcin Ozsoy, Ruihai Dong, Neil Hurley, Barry Smyth, and Aonghus Lawlor. Dares: An asynchronous distributed recommender system using deep reinforcement learning. *IEEE Access*, 9:83340–83354, 2021.

[54] Chengshuai Shi, Cong Shen, and Jing Yang. Federated multi-armed bandits with personalization, 2021.

[55] Jonathon Shlens. A tutorial on principal component analysis, 2014.

[56] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[57] Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, Keith Rush, and Sushant Prakash. Federated reconstruction: Partially local federated learning, 2022.

[58] Leslie N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[59] Ben Tan, Bo Liu, Vincent Zheng, and Qiang Yang. A federated recommender system for online services. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 579–581, New York, NY, USA, 2020. Association for Computing Machinery.

[60] Muhammad Ammad ud din, Elena Ivannikova, Suleiman A. Khan, Were Oyomno, Qiang Fu, Kuan Eeik Tan, and Adrian Flanagan. Federated collaborative filtering for privacy-preserving personalized recommendation system, 2019.

[61] Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16. ACM, September 2016.

[62] Shuai Wang, Richard Cornelius Suwandi, and Tsung-Hui Chang. Demystifying model averaging for communication-efficient federated matrix factorization. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3680–3684, 2021.

[63] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[64] Hongyi Zhang, Jan Bosch, and Helena Holmström Olsson. Federated learning systems: Architecture alternatives. In *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, pages 385–394, 2020.

[65] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. *Proceedings of the Twenty-first International Conference on Machine Learning*, pages 919–926, 2004.

[66] Yu Zhang, Christopher DuBois, and Michael D Ekstrand. A study of recommender systems for moocs. In *Proceedings of the 6th international conference on Computer supported education*, 2013.

[67] Pan Zhou, Kehao Wang, Linke Guo, Shimin Gong, and Bolong Zheng. A privacy-preserving distributed contextual federated online learning framework with big data support in social recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 33(3):824–838, 2021.