UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CAROLINA OLTRAMARI NERY

FIP-SHA - Um Método para Descoberta de Perfis Individuais através de Contas Compartilhadas

Dissertação apresentada como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação

Orientador: Prof. Dr. Weverton Cordeiro
Co-orientador: Prof. Dra. Renata Galante

Porto Alegre
2021

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Nery, Carolina Oltramari

FIP-SHA - Um Método para Descoberta de Perfis Individuais através de Contas Compartilhadas / Carolina Oltramari Nery. – Porto Alegre: PPGC da UFRGS, 2021.

78 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2021. Orientador: Weverton Cordeiro; Co-orientador: Renata Galante.

1. Contas Compartilhadas. 2. Agrupamento. 3. Similaridade entre Itens. 4. Perfil do Usuário. I. Cordeiro, Weverton. II. Galante, Renata. III. Título.

“Se podes olhar, vê. Se podes ver, repara.”

— LIVRO DOS CONSELHOS, JOSÉ SARAMAGO

AGRADECIMENTOS

Agradeço primeiramente á Universidade Federal do Rio Grande do Sul que me permitiu esses anos de formação acadêmica diferenciada e de qualidade. Agradeço ao professor Weverton Cordeiro pelo incentivo e ter me escolhido para entrar no mestrado e por ter me mostrado que existe a pesquisa, um mundo além das teorias.

Agradeço ao meu orientador Weverton Cordeiro e á minha co-orientadora Renata Galante por estarem comigo nesse estudo que gerou muitas discussões ciêntificas, me colocando em teste durante diversos momentos. Foram momentos de incerteza e insegurança, pensei em desistir, mas que resultaram nesse trabalho que têm por âmbito contribuir para o meio acadêmico. Obrigada pela paciência, por terem lido inúmeras vezes os mesmos capítulos e por terem me oportunizado essa experiência inesquecível.

Agradeço aos meus colegas de mestrado, Francielle Medeiros e Vanessa Borba, pela parceria nos trabalhos. Meus amigos Fábio Malet, Julia Vasconcellos, Juliana Sartori e Leonardo Faitão por ouvirem tantos desabafos, e o Nei Barbosa, cuja inteligência me fascina.

Agradeço aos meus familiares por todo apoio, principalmente á minha avó Lourdes e minha mãe Denise por sempre demonstrarem apoio e incentivo aos meus estudos, também agradeço minha irmã Gabriela e meu irmão Lucas, por serem grandes exemplos para mim. Minha mãe Denise, sempre foi sinônimo de luta e agradeço-a por estar junto a mim nos dias mais difícies, me apoiando e dando o seu máximo pela minha felicidade. Há, também preciso agradecer o meu cachorro Sushi, por todo o carinho.

Ás minhas tias Lucinara e Maria Tereza pelo incentivo e constante motivação que sempre me transmitiram.

Por fim, agradeço a banca avaliadora por aceitar o convite.

Obrigada,

RESUMO

Os sistemas de recomendação dependem do histórico de contas dos usuários (como itens visitados/comprados/classificados) para prever quais outros itens eles podem ter interesse. Na prática, várias pessoas (por exemplo, membros da família ou amigos) podem compartilhar uma única conta. Por esse motivo, extrair um único perfil de usuário a partir do histórico de uma conta pode levar a imprecisões nas sugestões de itens. O objetivo deste trabalho é propor um método para automaticamente descobrir os perfis de usuários presentes em contas compartilhadas. FIP-SHA, é o método proposto, cuja sigla é um acrônimo para *Finding Individual Profiles through SHared Accounts*. O método FIP-SHA está dividido em 3 etapas: (i) a quebra de sessão, quando se encontra um comportamento diferente do esperado, através do uso de similaridade de itens; (ii) representação das sessões; e (iii) a agrupamento das sessões que, em seu conjunto, representam cada perfil de usuário presente em uma conta. O FIP-SHA foi avaliado através de um conjunto de experimentos que, incialmente, avalia cada etapa do método, sendo o último experimento responsável por avaliar o resultado geral do agrupamento de sessões que representa os perfis dos usuários em uma conta compartilhada. Para realização dos experimentos, foram utilizadas duas bases de dados reais. Em comparação com o estado da arte, FIP-SHA mostrou-se eficaz para identificar a similaridade dos itens do usuário para a quebra das sessões online e o uso do método de agrupamento para agrupar essas sessões em perfis de usuário.

Palavras-chave: Contas Compartilhadas. Agrupamento. Similaridade entre Itens. Perfil do Usuário.

FIP-SHA - Finding Individual Profiles through SHared Accounts

ABSTRACT

Recommendation systems rely on users' account history (like visited/purchased/rated items) to predict which other items one could also be interested in. In practice, multiple individuals (e.g. family members or friends) may share a single account. For this reason, learning a single user profile from one's entire account history may lead to imprecise item suggestions. The main goal of this work is to propose a method to automatically discover the user profiles present in shared accounts. FIP-SHA is the proposed method, whose acronym is Finding Individual Profiles through SHared Accounts. The FIP-SHA is divided into 3 steps: (i) the break of the session, when a different behavior is found, through the use of similarity of items; (ii) representation of those sessions; and, (iii) the grouping of sessions that, together, represent each user profiles in an account; The FIP-SHA was evaluated through a set of experiments that, initially, evaluate each step of the method, with the last experiment being responsible for evaluating the general result of the grouping of sessions representing the profiles of users in a shared account. To carry out the experiments, two real databases were used. In comparison with the state of the art, FIP-SHA proved to be effective in identifying the similarity of user items for breaking online sessions and using the grouping method to group these sessions into user profiles.

Keywords: Shared Accounts, Clustering, Item-item Similarity, User Profile.

LISTA DE ABREVIATURAS E SIGLAS

AP	*Affinity Propagation*
FIP-SHA	*Finding Individual Profiles through SHared Accounts*
MAE	*Mean Absolute Error*
MAF	*Macro F-Score*
MIF	*Micro F-Score*
NMF	*Non-Negative Matrix Factorization*
NMI	*Normalized Mutual Information*
PCA	*Principal Component Analysis*
RMSE	*Root-mean-square Error*
SHE-UI	*Session-based Heterogeneous graph Embedding for User Identification*
SR	Sistemas de Recomendação
t-SNE	*t-Distributed Stochastic Neighbor Embedding*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Os Sistemas de Recomendação (SR) são aplicações de *softwares* que dão suporte aos usuários na busca por itens de interesse em um conjunto de objetos, geralmente de maneira personalizada. Atualmente, esses sistemas são usados em vários domínios de aplicativos, incluindo, por exemplo, *e-commerce* ou *streaming* de mídia (música ou filmes), e a recomendação personalizada tornou-se parte da experiência diária do usuário que navega *online*. Internamente, esses sistemas analisam o histórico dos usuários individuais ou de um conjunto de usuários como um todo para detectar padrões nos dados. Em plataformas *online*, vários tipos de ações relevantes de um usuário podem ser registradas, por exemplo, um item que o usuário favoritou ou uma compra realizada, e várias das ações de um único usuário podem estar relacionadas ao mesmo item. Essas ações registradas e os padrões detectados são usados para calcular recomendações que correspondem aos perfis de preferência de usuários individuais.

Na prática, no entanto, várias pessoas (por exemplo, familiares ou amigos) podem compartilhar uma única conta nessas plataformas. Por exemplo, vários membros de uma família podem compartilhar uma única conta na Amazon.com, pois é mais simples gerenciar todas as compras no mesmo cartão bancário (já salvo no sistema) e entregar essas compras em um único endereço familiar. O uso de uma única conta por várias pessoas representa um desafio ao fornecer recomendações personalizadas precisas. Informalmente, as recomendações fornecidas a uma conta "compartilhada", compreendendo as classificações de dois usuários diferentes, podem não corresponder aos interesses de qualquer um desses usuários. Além disso, o histórico dessa conta conterá ações de um grupo de indivíduos, em vez de uma única pessoa.

Com isso, os sistemas de recomendação que consideram o histórico de atividades dos usuários na plataforma podem gerar sugestões irrelevantes quando calculadas com base no histórico de toda a conta, sem ser aplicado um filtro. Considere, por exemplo, uma conta compartilhada em uma plataforma de *streaming* de vídeo onde uma pessoa assiste vídeos relacionados a esportes e a outra pessoa assiste vídeos sobre comédia. Nesse caso, as recomendações resultantes serão uma mistura de vídeos de esporte e comédia ou algum outro item que possa ser irrelevante para qualquer pessoa. Mais importante, a recomendação pode não capturar os interesses reais da pessoa que navega em um determinado instante, trazendo recomendações irrelevantes.

Empresas globais como Netflix e Amazon compartilham da mesma problemática.

Embora a Netflix permita que os membros criem até 5 perfis diferentes para cada conta com capacidade de personalização da experiência em cada perfil, uma grande porcentagem de perfis ainda é usada por várias pessoas em uma mesma casa. Embora os sistemas de recomendação tenham evoluido para fornecer uma mistura (união) de sugestões necessárias para realizar boas sugestões para qualquer membro da família que esteja visualizando a qualquer momento, essas visualizações em um grupo não são tão eficazes quanto as visualizações individuais. A Netflix (GOMEZ-URIBE; HUNT, 2015) aponta que ainda há muitas pesquisas e exploração para entender como trabalhar com os dados de visualização quando mais de uma pessoa estiver em uma sessão e criar recomendações para o cruzamento de dois ou mais interesses dos indivíduos em vez da união, como é feito atualmente. A Amazon.com (LINDEN; SMITH; YORK, 2003) apresentou um caso similar à Netflix, no qual são utilizados algoritmos de recomendação para personalizar as compras *online* para cada cliente. Assim, busca-se encontrar itens similares e não clientes similares, para cada compra.

Algumas empresas promovem competições para melhorar a precisão das recomendações, como o Prêmio Netflix (BELL; KOREN, 2007) e o desafio anual do RecSys[1]. No entanto, esses desafios frequentemente assumem que as ações associadas às contas de usuário refletem interesses individuais, o que não é o caso de uma conta compartilhada. Em uma palestra, Rastogi (RASTOGI, 2015) mencionou que lidar com várias pessoas por trás de contas individuais de clientes é um dos desafios de pesquisa enfrentados pela Amazon.com. Embora eles tenham progredido na previsão dos tamanhos de sapatos preferidos dos clientes (SEMBIUM et al., 2018), revelar os perfis dessas pessoas (incluindo interesses e preferências de mídia, compras e/ou notícias) continua sendo um problema em aberto.

O desenho de uma solução genérica para identificar os usuários presentes em uma conta compartilhada resulta na melhoria e facilidade na recomendação de produtos de forma mais eficaz. O desafio de pesquisa deste trabalho é responder a seguinte questão de pesquisa: "como descobrir diversos perfis de usuários presentes em uma conta compartilhada?". O objetivo deste trabalho é propor um método para automaticamente descobrir os perfis de usuários presentes em contas compartilhadas. FIP-SHA, é o método proposto, cuja sigla é um acrônimo para *Finding Individual Profiles through SHared Accounts*. O método FIP-SHA está dividido em 3 etapas, sendo elas: (i) a quebra de uma sessão quando encontramos um comportamento diferente do esperado, a análise é feita

[1]RecSys challenge Website: <http://www.recsyschallenge.com/2019/>.

calculando a similaridade entre os itens presentes em cada sessão. (ii) A representação das sessões através de vetores de termos e as frequências que descrevem os itens visitados. Por último, (iii) a terceira parte é representada pelo agrupamento das sessões que, em seu conjunto, representam cada perfil de usuário presente em uma conta.

O FIP-SHA foi avaliado através de um conjunto de experimentos que, incialmente, avalia cada etapa do método, sendo o último experimento responsável por avaliar o resultado geral do agrupamento de sessões que representa os perfis dos usuários em uma conta compartilhada. Para realização dos experimentos, foram utilizadas duas bases de dados reais. A primeira, Globo[2], armazena *logs* de interações de usuários do portal de notícias G1. A segunda,Lastfm[3], armazena sessões de escuta de músicas da plataforma Lastfm. Em comparação com o estado da arte, SHE-UI (JIANG et al., 2018), FIP-SHA mostrou-se eficaz para identificar a similaridade dos itens do usuário para a quebra das sessões online e o uso do método de agrupamento para agrupar essas sessões em perfis de usuário.

O restante do documento está organizado como segue. O Capítulo 2 apresenta os principais conceitos que embasam este trabalho. O Capítulo 3 apresenta uma revisão do estado da arte. No Capítulo 4 especifica o método FIP-SHA para descoberta de usuários em contas compartilhadas. Por fim, os Capítulo 5 apresenta os Experimentos enquanto o Capítulo 6 apresenta as conclusões e trabalhos futuros.

[2]<https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom>
[3]<http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html>

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda os conceitos necessários para o entendimento deste trabalho. Inicialmente, são definidos os conceitos de sistemas de recomendação, similaridade e redução de dimensionalidade. Por fim, o conceito de agrupamento e o algoritmo de *Affinity Propagation* que é utilizado com base para o agrupamento do método FIP-SHA proposto.

2.1 Sistemas de Recomendação

Sistemas de Recomendação são sistemas que devem ser capazes de predizer quais itens são relevantes para um usuário baseado nas preferências do mesmo (RICCI et al., 2010). Itens é o termo designado para se referir às recomendações (por exemplo, produtos em um site de *e-commerce*, vídeos e músicas). Já as preferências de um usuário podem ser classificadas como explícitas (por exemplo, avaliação de um produto) ou implícitas, inferidas de alguma interação do usuário com o item (por exemplo, a compra de um produto, a visualização de uma notícia ou a escuta de uma música). Uma sessão é uma sequência de requisições de um usuário para a aplicação em um determinado período de tempo (ARLITT, 2000). O tempo de uma sessão é definido pela sua aplicação, podendo ser considerado o tempo de inatividade do usuário o delimitador de corte, a diferença entre os itens em que o usuário interagiu, entre outras.

Uma sessão é classificada como um conjunto de itens (por exemplo, referindo-se a objetos, produtos, músicas ou filmes) que são coletados ou consumidos a partir de um evento (por exemplo, uma transação) ou em um determinado período de tempo ou uma coleção de ações ou eventos (por exemplo, ouvindo uma música) que ocorreram em um período de tempo (por exemplo, 60 minutos). Tanto um conjunto de itens comprados em uma transação quanto uma lista de músicas ouvidas por um usuário em uma hora podem ser vistos como uma sessão. Além disso, as páginas da *web* em que um usuário clicou sucessivamente em uma hora também podem ser consideradas uma sessão.

Um *timeout* é determinado como o tempo entre duas atividades sucessivas e é usado como um limite de sessão, significando que excedeu um determinado limiar estabelecido pela plataforma que está operando. Os seguintes eventos que podem determinar o final de uma sessão são (a) quando um usuário fecha o navegador da *web* ou efetua a saída (*logout*) ou (b) quando um usuário não realiza nenhuma ação por mais de um período de tempo definido arbitrariamente, a maioria das plataformas trabalham com essa

modalidade e estabelecem o tempo de 30 minutos.

2.2 Similaridade

Personalização, filtragem de informações e recomendação são técnicas fundamentais que ajudam os clientes que navegam *online* a se orientarem nas plataformas. A similaridade é um conceito importante no contexto de recomendação, pois o resultado depende do mecanismo de representação dos itens e o conjunto de informações. A semelhança é normalmente avaliada a partir das interações do usuário - a probabilidade de dois itens interagirem entre si no passado.

Para isso, é preciso entender as diferentes abordagens que existem, baseadas no usuário e as baseadas em itens que o usuário interagiu (NEBEL et al., 2017). A abordagem com base no usuário prevê o interesse de um usuário de teste em um item com base nas informações de classificação de perfis de usuários semelhantes. Cada perfil de usuário é classificado por sua diferença em relação ao outro perfil de usuário. As avaliações de usuários mais semelhantes contribuem quando se quer recomendar produtos semelhantes que um usuário teve interesse e que possivelmente o outro terá também.

Já a abordagem baseada em itens, aplica a mesma ideia, mas usa a similaridade entre itens em vez de usuários, e é evidenciada pelas interações do usuário, como avaliações e compras. Pode ser prevista pela média das classificações de outros itens semelhantes avaliados pelo usuário em questão. Cada item é classificado e reindexado de acordo com sua semelhança em relação a matriz de itens e, as classificações de itens mais semelhantes são ponderadas mais fortemente. Geralmente, utiliza-se a medida do Cosseno ou correlação de *Pearson* para o cálculo. A similaridade adotada para este trabalho é a baseada em itens e entraremos em maiores detalhes a seguir.

2.2.1 Similaridade Item-Item

A similaridade entre itens é estabelecida, principalmente, entre dois documentos (a e b) que utilizam uma representação de conteúdo (Ponnam et al., 2016), geralmente representado por um vetor de termos/palavras, e realiza uma comparação entre os dois conteúdos. Ao final do cálculo de similaridade, é possível estabelecer um *ranking* entre quais itens são mais, e os que são menos similares às preferências dos usuários e utilizar

esta informação para realizar a tarefa de recomendação desejada. Uma das abordagens mais comumente utilizadas é a similaridade por Cosseno (MUSA; ZHIHONG, 2020), em que a similariade entre duas sessões s_1 e s_2 é definida pela fórmula:

$$cos(s_1, s_2) = \frac{s_1 \cdot s_2}{\|s_1\| \, \|s_2\|} \tag{2.1}$$

Produto ($\cdot$) escalar entre dois vetores.

2.3 Word Embeddings

Word Embeddings são representações numéricas de texto em que os números usados para representar palavras individuais não indicam apenas a presença ou ausência de uma palavra ou frequência, mas capturam o significado e as relações semânticas dessa palavra com as palavras em seu contexto. Portanto, as *Word Embeddings* capturam a semelhança semântica entre os *tokens* ou *pixels* e os projetam no espaço vetorial definido pelo usuário (LI et al., 2015). Pode-se dizer que *Word Embeddings* são espaços de baixa dimensão que podem projetar um vetor de alta dimensão. Pensando em um contexto, por exemplo, trabalhar com uma imagem ou palavra que pode ter milhões de parâmetros caracterizados por *pixels* ou *tokens*, é preciso de uma estrutura uniforme que seja a entrada para os modelos de aprendizado de máquina.

Existem várias técnicas que podem ser usadas para converter palavras em números, como *one-hot*, *Count Vectorizer* e *Prediction*. No contexto do trabalho, quando dimensões altas ou um grande número de recursos são modelados para derivar um padrão em tamanho de dados limitado, geralmente encontra-se o problema de dimensionalidade, ou seja, o modelo é incapaz de extrair padrões relevantes dos dados de entrada. Por isso, *Principal Component Analysis* (PCA) e *T-Distributed Stochastic Neighbor Embedding* (t-SNE) são algumas das técnicas utilizadas para reduzir a dimensionalidade de espaços de vetores de palavras e visualizar as *Word Embeddings* e agrupamentos (*clusters*) de palavras (MAATEN; HINTON, 2008a; KAUFMAN; ROUSSEEUW, 1990). As *Word Embeddings* representam a entrada para o método t-SNE, e a sua saida é a representação dessas palavras em um espaço de dimensão 2D.

2.4 Padrões através de Redução de Dimensionalidade

A redução de dimensionalidade é frequentemente empregada para mapear dados dimensionais elevados para um espaço dimensional inferior, e manter o máximo de informações possíveis (LU; PLATANIOTIS; VENETSANOPOULOS, 2011; JOLLIFFE, 1986; JOLLIFFE, 2011). A seguir, são descritos os dois algoritmos utiizados neste trabalho.

2.4.1 PCA

Um dos algoritmos mais conhecidos é o *Principal Component Analysis* (PCA) (HOTELLING, 1933), e seu primeiro relato presente na literatura foi utilizado para separar um conjunto de características de dados na área da psicologia. PCA consiste em pegar as N dimensões das características do conjunto de dados, e produzir novos N eixos que são combinações lineares que maximizam o desvio padrão dos pontos desse espaço, após isso, a técnica de *scree-plot* é aplicada, resultando em um *ranking* dessas combinações lineares pelo seu desvio padrão, assim, escolhe-se os dois eixos com mais variabilidade de dados, e usando-os é possível gerar o plano cartesiano representando o corte nesse espaço multidimensional que possui maior relevância e sensitividade no conjunto.

No entanto, PCA apresenta limitações na sua linearidade (MAMMO; LINDGREN, 2020). Uma desvantagem da técnica PCA é a sua projeção linear, significando que não consegue capturar dependências não lineares. Por exemplo, em um conjunto de dados multidimensional, o plano que melhor representa a sensitividade do conjunto de dados é um plano não-euclidiano (o plano curvo), nesse caso encontraríamos um corte não ótimo em termos de representatividade deste espaço.

2.4.2 t-SNE

Para encontrar os planos não-euclidianos em um espaço N-dimensional, *t-Distributed Stochastic Neighbor Embedding* (t-SNE), (MAATEN; HINTON, 2008a), utilizam-se técnicas de *Machine Learning* para, em um conjunto de etapas, convergir ao plano 2D que melhor representa o conjunto de características que produzem maior relevância ao conjunto de dados. O objetivo é manter os pontos semelhantes juntos e pontos diferentes

separados. Normalmente, a distância euclidiana entre os pontos é usada como uma medida de similaridade.

O método t-SNE não se limita às projeções lineares (SAKIB; SIDDIQUE; RAHMAN, 2020), o que o torna adequado para todos os tipos de conjuntos de dados. Suas principais características são: (a) usa os relacionamentos locais entre os pontos para criar um mapeamento de baixa dimensão, isso o permite capturar uma estrutura não linear. (b) A distribuição de probabilidade é criada usando a distribuição Gaussiana que define as relações entre os pontos no espaço de alta dimensão. (c) t-SNE usa a distribuição *Student t-distribution* para recriar a distribuição de probabilidade no espaço de baixa dimensão evitando o problema de aglomeração. E, (d) t-SNE otimiza os encaixes diretamente usando o gradiente descendente.

Sua implementação é baseada na minimização da divergência entre duas distribuições: uma distribuição mede semelhanças entre pares dos objetos de entrada e uma distribuição que mede semelhanças entre pares dos pontos de baixa dimensão correspondentes. O t-SNE, descrito em um alto nível, funciona da seguinte maneira: na primeira etapa, ainda no espaço de alta dimensão, cria uma distribuição de probabilidade que dita os relacionamentos entre vários pontos vizinhos (HINTON; ROWEIS, 2003). Na segunda etapa, o t-SNE tenta recriar um espaço de baixa dimensão que segue essa distribuição de probabilidade da melhor maneira possível. O t do t-SNE representa a distribuição t, que é a distribuição usada na segunda etapa. O S e o N são de estocástico e vizinho, e provêm do fato de usar uma distribuição de probabilidade entre pontos vizinhos.

Portanto, pode-se dizer que principais diferenças entre PCA e t-SNE são a redução de dimensionalidade de linearidade versus não linearidade e a função objetivo, ou seja, PCA busca preservar a estrutura de dados global, enquanto o algoritmo t-SNE preserva a estrutura local.

2.5 Agrupamento

Agrupamento é um método não supervisionado que visa descobrir um agrupamento ideal (que tenham algum significado) em uma coleção (TAN et al., 2018). A similaridade dos itens é determinada usando funções de distância (Jia Rongfei; Jin Maozhong; Liu Chao, 2010; BOBADILLA et al., 2013) e os algoritmos de agrupamentos buscam minimizar a distância intra-grupos e maximizar a distância inter-grupos (RICCI et al., 2010).

Como o objetivo deste trabalho é agrupar as sessões de uma conta compartilhada

pelas suas similaridades, a técnica escolhida foi realizar o uso de agrupamentos, onde cada agrupamento contém sessões e está associado a um perfil de usuário.

2.5.1 Algoritmos

O *k-means* é um algoritmo clássico de agrupamento, simples e eficiente, e amplamente utilizado em tarefas de agrupamento (RICCI et al., 2010). O algoritmo trabalha com centróides que são normalmente escolhidos aleatoriamente, onde os itens são atribuídos aos grupos mais próximos e os centróides são atualizados continuamente até não haver mais mudança de itens entre grupos (UNGAR; FOSTER, 1998).

Porém, existem algumas limitações conhecidas sobre o *k-means*. Pode-se citar alguns pontos: (i) conhecimento a *priori* dos dados, para a escolha do número *k* de grupos que será gerado; (ii) os grupos gerados no final são muito sensíveis à escolha inicial dos centróides; (iii) e pode produzir grupos vazios (RICCI et al., 2010).

Uma alternativa ao *k-means* é o algoritmo de agrupamento Propagação de Afinidades[1]. É um algoritmo proposto recentemente que ganhou grande popularidade na aplicação em áreas da bioinformática, apresentando bons resultados para problemas de agrupamentos de sequência de DNA, mas também vem sendo aplicado em outras áreas, como agrupamento de faces (imagem) (BODENHOFER; KOTHMEIER; HOCHREITER, 2011), combinado a outros métodos em coleções de filmes (AMATRIAIN, 2013), e na sumarização de textos (RICCI et al., 2010).

2.5.2 Affinity Propagation - Agrupamento através da passagem de mensagens

O método *Affinity Propagation* (FREY; DUECK, 2007) encontra o número de agrupamentos automaticamente, sendo estes representados pelo elemento que melhor generaliza todos os elementos dentro do grupo. Baseia-se na troca de mensagens entre itens até que sejam encontrados exemplares que representam cada agrupamento. Recebe como entrada as similaridades entre cada item e a cada iteração são passadas dois tipos de mensagens: as responsabilidades e as disponibilidades, que são calculadas de acordo com as similaridades.

A maioria dos algoritmos de agrupamento usam como parâmetro de entrada, um

[1] do inglês *Affinity Propagation* (AP)

número pré-determinado k de agrupametos para particionar o espaço amostral de dados. O *Affinity Propagation* (AP) adota o princípio de que todos os pontos de dados podem ser eleitos como "exemplar do agrupamento". O conjunto de dados forma uma configuração de rede em que os pontos representam os nós e as transmissões de mensagens ocorrem entre as arestas da rede. Recebe como entrada uma coleção de semelhanças com valores reais entre pontos de dados, onde a semelhança $s(i, k)$ indica quão bem o ponto k é adequado para ser o exemplo do ponto de dado i. Os centróides retornados do algoritmo são pontos de dados reais e esses pontos de dados são a base para o que se define como um ponto de dado relevante em um conjunto de dados.

O objetivo do *Affinity Propagation* é descobrir os exemplos através de um processo de transmissão de mensagens. Esse algoritmo de transmissão de mensagens é baseado no algoritmo *Sum Product* (DUECK, 2009) e visa encontrar um valor máximo de responsabilidade e disponibilidade de cada ponto de dado. A responsabilidade é a medição da capacidade de um ponto de dados ser atribuída a um agrupamento e a disponibilidade é a medida da capacidade de um ponto de dados ser rotulado como um exemplo para os pontos de dados atribuídos a ele.

Assim, o algoritmo é executado iterativamente através de cada ponto de dados, passando mensagens e escolhendo exemplos até que a convergência seja alcançada, significando que as disponibilidades e responsabilidades de cada ponto de dados não são mais atualizadas. Portanto, a cada iteração surgem novos exemplos e agrupamentos serão formados. Quando a convergência é alcançada, um conjunto de exemplos é selecionado e os agrupamentos assumem sua forma final. A precisão também é muito importante no algoritmo de agrupamento, pois os centróides precisam representar o mais próximo possível das atribuições de dados de um agrupamento.

Uma das propostas desse trabalho é utilizar o algoritmo *Affinity Propagation* na descoberta de perfis de usuários em contas compartilhadas, com o intuito de investigar se os bons resultados que o algoritmo tem mostrado são válidos, realizando a comparação entre bases de dados diferentes.

2.5.3 Affinity Propagation - o algoritmo

Affinity Propagation tem basicamente uma entrada, os valores das similaridades $s\,(i,\ k)$ entre cada par de pontos $\{x_i,\ x_k\}$, estas semelhanças indicam o quão o ponto x_k seria apto para representar o ponto x_i. Valores altos de preferência farão com que o

Affinity Propagation encontre muitos agrupamentos, enquanto valores baixos levarão a um pequeno número de agrupamentos. Uma boa opção inicial para determinar a preferência é obter as semelhanças mínimas ou medianas. A semelhança é comumente expressa como uma distância euclidiana negativa ao quadrado de acordo com a equação 2.2, na qual os parâmetros x_i e x_j são as posições dos pontos de dados i e j no espaço $2D$.

$$s(i,k) = -\left\|x_i - x_k\right\|^2 \tag{2.2}$$

Além das similaridades, *Affinity Propagation* também tem como parâmetro de entrada valores $s\left(k,\ k\right)$ denominados "preferências" para cada ponto k. Sabendo, que em qualquer métrica a diagonal principal da matriz de similaridade é nula, isso indica que a maior similaridade ocorre de um ponto a ele mesmo, ou seja, as preferências para esses pontos são exemplares. Dito isso, os valores de preferência mencionados interferem nos resultados (número de agrupamentos) encontrados pelo algoritmo, altos e baixos valores de $s\left(k,\ k\right)$ resultam em grandes e baixos números de agrupamentos encontrados, respectivamente.

A troca de mensagens entre os pontos podem ser de dois tipos, transmissão de responsabilidades e disponibilidades. As responsabilidades $r\left(i,\ k\right)$, mensagens enviadas de um ponto i para um possível exemplar k indicam o quão adequado seria o ponto k ser exemplar para o ponto i. As disponibilidades $a\left(i,\ k\right)$, enviadas do candidato exemplar k para o ponto i, indicam evidências de quão adequado seria para o ponto i escolher o ponto k como seu exemplar e são iniciadas com zero.

As responsabilidades são definidas pela regra representada pela Equação 2.3:

$$r\left(i,\ k\right) \leftarrow s\left(i,\ k\right) - \underset{k' \neq k}{max}\left\{a\left(i,\ k'\right) + s\left(i,\ k'\right)\right\} \tag{2.3}$$

Na equação 2.3, i representa um ponto e k' representa um exemplo de candidato concorrente. Na primeira iteração, como as disponibilidades são inicializadas como zero, $r(i,k)$ é definido como a semelhança de entrada entre o ponto i e o ponto k como seu exemplar, menos o máximo das semelhanças entre o ponto i e outros exemplos candidados k'.

As disponibilidades, que definem se um exemplar é bom ou não, são definidas pela

Equação 2.4:

$$\begin{cases} a\left(i,\ k\right) \leftarrow min\left\{0, r(k,k) + \sum\limits_{i' \notin \{i,k\}} max\left\{0,\ r(i',k)\right\}\right\}, \\ a(k,k) \leftarrow \sum\limits_{i' \neq k} max\left\{0,\ r(i',k)\right\} \end{cases} \tag{2.4}$$

Nas equações acima, a disponibilidade $a(i,k)$ é definida como a auto-responsabilidade $r(k,k)$ mais a soma das responsabilidades positivas que o candidato exemplar k recebe de outros pontos de apoio i'. Apenas as partes positivas das responsabilidades recebidas são adicionadas, Equação 2.5.

$$AP = max\left\{a(i,k) + r(i,k)\right\} \tag{2.5}$$

O processo de propagação de mensagens encerra assim que atinge um número especificado de iterações ou quando a estrutura do agrupamento se estabiliza com um determinado número de iterações (DUECK, 2009).

Esses pontos representam as diagonais das matrizes de disponibilidade e responsabilidade. A matriz de responsabilidade é usada para determinar quais pontos de dados são atribuídos a esses exemplos. É devido a esse procedimento final que o *Affinity Propagation* retorna um resultado "natural" de agrupamento, enquanto outros algoritmos inicializam o número de agrupamentos, conforme determinado pela auto-disponibilidade e auto-responsabilidade. É possível que o algoritmo não atinja a convergência, e isso significa que os valores de responsabilidade e disponibilidade ainda estão atualizando. Como resultado, as atribuições e exemplares podem diferir se o algoritmo for executado novamente.

Com o algoritmo *Affinity Propagation*, busca-se a convergência para qualquer conjunto de dados, os agrupamentos e os exemplos retornados são obtidos com influência do valor do fator de amortecimento (*damping*) que varia entre 0,5 e 1. Possui uma complexidade de tempo de $O\left(k * n^2\right)$, onde n é o número de registros e k representa o número de iterações (REFIANTI; MUTIARA; GUNAWAN, 2017).

3 TRABALHOS RELACIONADOS

Este capítulo apresenta o estado da arte em que o método FIP-SHA se insere. As áreas de pesquisa são apresentadas, identificando as principais necessidades e desafios no estudo da identificação de perfis em contas compartilhadas. Os principais trabalhos relacionados são descritos e um comparativo é realizado. Os trabalhos relacionados estão separados em duas seções: similaridade entre sessões, seção 3.1 e, identificação de usuários em contas compartilhadas, seção 3.2. O baseline do trabalho, SHE-UI está descrito na seção 3.3. Por fim, um comparativo com os principais trabalhos encontrados na literatura é apresentado em 3.4.

3.1 Similaridade

A recomendação baseada em conteúdo é fundamentada na similaridade dos itens recomendados. A ideia básica é que, se um usuário gosta de um determinado item, também gostará de um item semelhante e funciona bem quando é fácil determinar as propriedades de cada item – e logo, sua semelhança com outros itens, calculando matematicamente a importância dos termos utilizados para descrever os itens.

O método que está inserido o trabalho analisa os dados fornecidos pelos usuários, como as classificações de filmes, músicas e notícias. A seguir, os trabalhos relacionados estão descritos. Para realizar a quebra/identificação das sessões existe uma abordagem que utiliza heurísticas orientadas ao tempo para reconstruir sessões de usuários. Estudos avançaram e Xinhua (XINHUA; QIONG, 2011) apresenta um algoritmo de identificação de sessão baseado em tempo dinâmico. No início do algoritmo, o tempo limite é determinado para uma página da *Web* usando os resultados estatísticos, em combinação com o grau de importância de uma página. Depois que o procedimento de identificação da sessão é iniciado, o tempo limite é alterado dinamicamente. Os resultados dos experimentos mostram que o algoritmo proposto pode ter um desempenho melhor que os algoritmos baseados no tempo tradicional usado para a identificação da sessão.

No contexto de notícias, Sottocornola (SOTTOCORNOLA; SYMEONIDIS; ZANKER, 2018) trabalha com usuários anônimos ou registrados. Para usuários anônimos, não há um perfil de usuário e são armazenadas as sessões anônimas juntamente com os artigos acessados. Para os usuários registrados ou os que aceitaram solicitações de *cookies*, as interações no passado são acompanhadas e cria-se um perfil de usuário. São dois módu-

los, o primeiro é um atualizador de perfil e o segundo é o recomendador para entregar os *top-N* itens recomendados para cada usuário. O módulo atualizador de perfil lê instâncias das sessões do usuário, combinando-as com informações gravadas anteriormente sobre as principais entidades (usuários, itens e sessões). Em seguida, uma *sliding time* de tamanho w indica que o processamento em um ponto de tempo t deve considerar todos os eventos em menos de $t - w$. Portanto, o atualizador de perfil define um intervalo de validade $[t - w, t)$ no qual calcula-se a semelhança entre os itens com base nas interações entre artigos e sessões ou nas categorias de tópicos dos artigos. Em seguida, essas informações são fornecidas ao módulo de recomendação, para sugerir os *N* principais itens para cada usuário.

O método FIP-SHA, utiliza a similaridade para realizar o cálculo entre as sessões e identificar o início e o fim de cada sessão. Assim, quando observa-se um conteúdo diferente do perfil, realiza uma quebra na sessão corrente. Pesquisas na literatura, mostram-se incapazes da identificação do início e fim da sessão quando existe mais de um usuário que navega *online*, e a utilização de um delimitador de tempo (por exemplo, 30 minutos de inatividade) não apresenta ser um bom parâmetro para a realização do corte nas sessões.

3.2 Identificação de Usuários em Contas Compartilhadas

Esta seção apresenta os principais trabalhos que fazem parte do contexto de identificação de usuários em contas compartilhadas. Contas compartilhadas são contas que usam um único par de credenciais para autenticar vários usuários. Embora contas compartilhadas não sejam consideradas práticas recomendadas, uma organização pode acabar usando contas compartilhadas por diversos motivos.

Zhang et al. (ZHANG et al., 2012) foram os primeiros a estudar a identificação de usuários no contexto de plataformas de filmes e, as informações utilizadas são unicamente as classificações fornecidas pelos usuários. O objetivo dos autores é identificar se uma determinada conta é compartilhada e agrupar as ações dos usuários que a compartilham. Para a finalidade, desenvolveu-se um modelo para contas compartilhadas com base em uniões de subespaços lineares e agrupamento de subespaços aplicados para executar a tarefa de identificação; no trabalho de Zhang recomenda-se a união de itens com maior probabilidade de serem altamente classificados por cada usuário. Bajaj (BAJAJ; SHEKHAR, 2016) propõem um método de agrupamento de canais baseados em similaridade para agrupar canais semelhantes para as contas e usam o algoritmo *Apriori* para

decompor a conta de TV *online* em pessoas distintas que compartilham a conta através da análise das características de visualização e individualizar a experiência de cada pessoa. Depois disso, utiliza-se de perfis pessoais para recomendar canais adicionais à conta. Wang (WANG et al., 2014) supõe que diferentes usuários consumam serviços em diferentes períodos, para tanto, decompõem-se os usuários com base em diferentes preferências de mineração ao longo de diferentes períodos de tempo dos *logs* de consumo. Por fim, utiliza-se um método *user-KNN* para fazer recomendações para cada usuário identificado. Yang (YANG et al., 2015) também analisam a semelhança da proporção de cada tipo de item em um período de tempo para deduzir se uma sequência é gerada pelo mesmo usuário. Em seguida, recomendações são geradas para um usuário específico recomendando gêneros personalizados aos usuários identificados.

Verstrepen (VERSTREPEN; GOETHALS, 2015) apresenta um estudo das *top-N* recomendações para contas compartilhadas na ausência de informações contextuais, baseados em itens. Os dados são representados como uma matriz de preferência na qual as linhas representam os usuários e as colunas representam os itens. Todo valor nessa matriz de preferência possui valores 1 ou 0, o valor 1 representa uma preferência conhecida e o valor 0 representa a desconhecida. É referida como recomendação de $N - N$, com base em dados binários de preferência somente positivos, portanto, gostos em sites de redes sociais são explícitos, binários e apenas positivos. No artigo, os perfis individuais na conta compartilhada são desconhecidos e supõem-se que todo usuário na conta compartilhada possa identificar as recomendações destinadas a ele e consumir essas recomendações individualmente. Para um usuário u, esse sistema de recomendação encontra KNN(j), os k itens mais semelhantes a j, para cada item preferido j usando uma medida de similaridade $sim(j, i)$. Apesar disso, Verstrepen não identifica o perfil de cada usuário por trás das contas compartilhadas.

Yang (Yang et al., 2017) identifica usuários utilizando-se de um método não supervisionado baseado em projeção e, em seguida, faz uso de técnicas de *Factorization Machine* para prever a preferência de um usuário com base em informações históricas para gerar recomendações personalizadas. Jiang (JIANG et al., 2018), apresenta uma estrutura (SHE-UI) baseada em aprendizado não supervisionado para diferenciar as preferências dos usuários e as sessões de grupo por usuário no domínio de *streaming* multimídia. Uma estrutura baseada em aprendizado por *feature*, aprendizado não supervisionado e *normalized random walks* é utilizada e, os autores são capazes de identificar um conjunto de usuários por trás de uma conta compartilhada, a partir das sessões de *streaming* de mú-

sica. Além disso, dada uma nova sessão de *streaming* de uma conta, SHE-UI é capaz de corresponder a uma persona identificada. A solução possui dependência na extração de recursos e no uso da técnica *random walk* utilizando-se de itens unicamente homogêneos (como itens de música), limitando sua aplicabilidade no cenário de *e-commerce*.

Sembium (SEMBIUM et al., 2018) estuda a solução para o problema de recomendação de itens para usuários através de contas compartilhadas na Amazon.com no contexto da recomendação de tamanho de produto para itens como roupas e sapatos. O problema das recomendações de tamanho de produto com base nos dados de compras e devolução do cliente resulta em altas taxas de retorno pelo cliente comprar tamanhos incorretos, apresentando-se como um problema importante no domínio do comércio eletrônico, produtos como roupas e calçados continuam desafiando a compra *online* e taxas de retorno recorde. O que leva o cliente a retornar o produto é o problema do tamanho adequado e a escolha do tamanho correto, portanto, além da falta de experiência do usuário, existe a variabilidade e o dimensionamento do produto entre marcas e tipos de produto. No caso de contas compartilhadas, aprender um único tamanho real pode levar a estimativas imprecisas do tamanho verdadeiro. A abordagem que Sembium utiliza para lidar com várias *personas* é introduzir uma matriz de variáveis latentes em seu modelo para capturar várias *personas*. Além de ser específica à abordagem dos autores, a solução não lida com itens de natureza diversa.

π-Net (MA et al., 2019) estuda a recomendação sequencial entre domínios de conta compartilhada, no qual o comportamento do usuário em uma conta compartilhada é registrado em vários domínios. Nesse estudo entende-se por domínios, por exemplo, um domínio educacional e um domínio de vídeo, pois trata-se de dados de *Smart* TV. A ideia é que o comportamento do usuário em domínios diferentes possa refletir interesses semelhantes, além disso, os autores afirmam que um domínio pode ser útil para melhorar as recomendações em outros domínios. O sistema contém duas unidades principais: uma unidade de filtro de conta compartilhada (SFU) e uma unidade de transferência de domínio (CTU). CTU extrai e compartilha recorrentemente informações entre dois domínios (A e B, por exemplo) para melhorar o desempenho das recomendações para ambos os domínios e uma RNN é usada para codificar as sequências de comportamento de cada domínio em representações de alta dimensão. As saídas do codificador de sequências são representações de todos os usuários que compartilham a mesma conta, e para aprender as informações específicas do usuário a partir das representações mistas, existe o módulo SFU. Apesar da solução, (i) é assumido que existem K usuários em cada conta, ou seja,

é um valor pré-definido e, (ii) no estudo, utilizou-se 2 domínios (A e B), e os autores alegam que a solução é aplicável para mais domínios. Além disso, o método π-Net apresenta melhores resultados quando compartilham informações em dois domínios que se complementam. Quando há apenas um domínio ou os dados em dois domínios compartilham menos informações, o método π-Net não é tão eficaz. Como trabalho futuro, pretendem detectar automaticamente o número de membros nas contas, atualmente assume-se que o mesmo número de usuários está presente em todas as contas compartilhadas.

Pesquisas mais recentes demonstraram preocupação no compartilhamento de contas (LIN et al., 2020; OBADA-OBIEH; HUANG; BEZNOSOV, 2020). Lin realizou um estudo onde analisa, durante 30 dias, o uso diário entre casais que compartilham contas digitais por conveniência, e descobriu novos comportamentos. Obada-Obieh et al. (OBADA-OBIEH; HUANG; BEZNOSOV, 2020) sugerem melhorias de design de contas *online* para oferecer melhor suporte aos usuários quando encerram o compartilhamento em uma conta.

As preocupações com segurança e privacidade devido ao uso compartilhado de dispositivos em ambientes automatizados entre vários usuários em uma única casa têm recebido atenção. Porém ainda há um número limitado de pesquisas, nesse contexto Geeng et al. (GEENG; ROESNER, 2019) buscam descobrir quem está no comando em questão, e descobriram desequilibros de poder entre usuários primários e secundários. Huang et al. (HUANG; OBADA-OBIEH; BEZNOSOV, 2020) identificaram o controle de privacidade inadequado sobre os dispositivos inteligentes no uso doméstico.

3.3 Baseline

Como solução para o problema de contas compartilhadas em serviços de *streaming*, tais como Netflix e Spotify, os autores do *framework* SHE-UI (JIANG et al., 2018) propõem utilizar algoritmos de aprendizado não supervisionado e agrupamento para identificar usuários em contas compartilhadas e torna-se o *baseline* do método FIP-SHA. Embora tenha sido projetado originalmente para o contexto de músicas, os autores afirmam em seu artigo que pode ser utilizado em outros contextos, citando como exemplo plataformas de filmes.

3.3.1 SHE-UI

O *framework* SHE-UI visa identificar múltiplos usuários em uma conta compartilhada a partir de *logs* passados de requisição de músicas em cada conta. Estes *logs* mantém informações de quais músicas foram escutadas por uma conta junto com seu respectivo *timestamp*. A ideia do *framework* é encontrar diferentes padrões de itens requisitados em cada sessão e associar cada padrão a um usuário distinto.

O padrão de cada sessão é derivado a partir das *features* de cada música que ela contém. Estas *features* são aprendidas a partir das informações a respeito de cada música, são os metadados que compõem as características da música, tais como artista, álbum e gênero. Conforme aumentam as informações de cada música, melhor o *framework* consegue aprender suas *features* e portanto melhor tende ser a identificação dos usuários.

O processo de aprendizagem ocorre em três etapas. A primeira é a construção de um grafo que representa as relações entre as músicas e seus metadados, nele cada nodo representa um item ou um metadado e cada aresta representa a relação entre os nodos. A segunda etapa consiste em mapear o grafo para um espaço vetorial, onde cada música é representada por um vetor de *features*, ao invés de um nodo do grafo. Para isso, a arquitetura *Word2Vec* é utilizada, de forma que cada nodo do grafo é tratado como uma palavra e caminhamentos aleatórios são gerados para serem utilizados como frases. A terceira etapa trabalha o agrupamento das sessões de cada conta em diferentes grupos para a obtenção do número de usuários da conta compartilhada, cada um contendo as sessões de um único usuário. Esta tarefa é realizada explorando-se as informações contidas no espaço vetorial produzido na etapa anterior. Para esta tarefa, propõem-se um algorimo simplificado da versão original do algoritmo de *Affinity Propagation* (FREY; DUECK, 2007).

O algoritmo proposto para o *framework* é dito uma versão simplificada do *Affinity Propagation*, comparando com o algoritmo tradicional que utiliza um fator de amortecimento (*damping*) em suas regras de atualização para evitar oscilações numéricas na convergência, enquanto que o algoritmo proposto não faz uso desse parâmetro.

3.3.2 Resultados do Baseline

O *framework* SHE-UI utiliza dois conjuntos de dados diferentes para conduzir seus experimentos. Um destes contém informações sensíveis a respeito dos usuários e portanto

não pôde ser compartilhado. Já o outro conjunto disponibilizado é uma versão de um conjunto de dados público da plataforma Last.fm que passou por um pré-processamento. Este pré-processamento incluiu as tarefas de: (i) separação dos *logs* de cada conta em sessões, cada sessão separada por um período de inatividade de 30 minutos; (ii) remoção dos *timestamps*; (iii) e a remoção de músicas e sessões com menos de 10 requisições.

São dois os objetivos definidos pelo *framework*: *UI-Past* e *UI-New*. *UI-Past* identifica usuários a partir do *log* de sessões passadas, enquanto que *UI-New* identifica o usuário que navega em uma sessão atual. As tabelas 3.1 e 3.2 mostram os resultados:

Tabela 3.1: Resultados para número conhecido de usuários

	UI-Past			*UI-New*		
Métrica	NMI	MAF	MIF	NMI	MAF	MIF
Original	0.6108	0.7613	0.8393	0.5718	0.7455	0.8236
Implementado	0.6273	0.7228	0.8276	0.5626	0.7025	0.7947

Tabela 3.2: Resultados para número desconhecido de usuários

	UI-Past			*UI-New*		
Métrica	NMI	MAF	MIF	NMI	MAF	MIF
Original	0.3375	0.6563	0.6782	0.3214	0.6323	0.6568
Implementado	0.2540	0.4653	0.5977	0.2223	0.4158	0.5348

Os resultados obtidos pelo segundo experimento, Tabela 3.2, apresentaram-se ruins e mostram um erro no código, o motivo da desconfiança justifica-se porque o resultado obtido a partir do algoritmo de agrupamento não consegue encontrar nenhum agrupamento, ou seja, o número de usuários retornado é zero.

O artigo que apresenta o método SHE-UI, contém dúvidas referentes à implementação, detalhes técnicos são omitidos e o código não está disponível publicamente. Além disso, não ficou claro como cada métrica foi calculada, apenas quais métricas foram utilizadas. Os resultados para cada número de usuários em uma conta compartilhadas são proporcionais ao erro, quanto maior o número de usuários compartilhando uma conta, maior é a margem de erro do *framework*. Por fim, é testado somente um tipo de cenário (músicas), alega-se que outros cenários se adequariam ao uso do *framework*.

3.4 Comparativo

Esta seção apresenta um compartivo entre os principais trabalhos encontrados na literatura e que se enquadram no estudo da identificação de perfis no contexto de contas compartilhadas.

Critérios como a identificação do usuário que navega *online* é utilizado para a comparação, nesse cenário os trabalhos de Jiang, Bajaj e Yang (JIANG et al., 2018; BAJAJ; SHEKHAR, 2016; YANG et al., 2015) conseguem realizam a identificação do usuário. O segundo critério utilizado para o comparativo é a métodologia utilizada para a identificação dos perfis, Jiang trabalha no domínio de *streaming* multimídia e a solução dos autores depende fortemente da extração de recursos e da caminhada aleatória (*normalized random walks*) em itens homogêneos (como itens de música), o que limita sua aplicabilidade em um cenário de *e-commerce*. Bajaj analisa os padrões de visualização dominantes em cada conta e é calculada a similaridade de Cosseno entre eles, assim é criada uma Matriz de Frequência que conta os vídeos assistidos em cada conta por cada canal, e são agrupados em grupos com base nesses valores de similaridade. Yang fez uso de informações contextuais (por exemplo, dispositivo, local, hora) presentes nos *logs* de visualização do usuário. Zhang (ZHANG et al., 2012) apresenta uma abordagem que é limitada à classificação no cenário de filmes, agrupa os eventos de classificação (*ratings*) e juntamente com o algoritmo EM (*Expectation Maximization*) identifica o perfil que melhor prevê cada classificação.

O terceiro critério, se consegue trabalhar com diversos cenários, um escopo de músicas e notícias, por exemplo. Por fim, o quarto critério, é a utilização de um método de agrupamento. Verstrepen (VERSTREPEN; GOETHALS, 2015) não descobre os perfis em uma conta compartilhada e opta por deixar com que o usuário reconheça qual recomendação se destina a ele, assim cada usuário pode se identificar com suas preferências, encontra os KNN(j) k itens mais semelhantes a j, para cada item preferido utilizando a similaridade por Cosseno. (MA et al., 2019) trabalha com uma rede RNN em *logs* de canais de televisão e não consegue identificar os usuários em uma conta compartilhada.

A Tabela 3.3 realiza o comparativo entre os trabalhos mais recentes encontrados no estado da arte, os métodos não comprovam a aplicação diretamente na identificação de perfis em contas compartilhadas visando multiplos cenários.

Tabela 3.3: Informações das bases de dados utilizados.

	Identifica os perfis dos usuários por trás de contas compartilhadas?	Método para identificação	Escopo	Método de Agrupamento
Zhang et al.	Sim	Agrupa os eventos de classificação	Filmes	*K-means* e *spectral clustering*
Verstrepen et al.	Não	O usuário identifica as suas preferências	Filmes	*KNN*
Jiang et al.	Sim	Grafo de metadados e Agrupamento	Músicas	*Affinity Propagation*
Bajaj et al.	Sim	Similaridade Cosseno	Canais de TV assistidos	Agrupamento hierárquico
Yang et al.	Sim	PCA e uso de informações do dispositivo, local e hora nos *logs*	IPTV (vídeos)	Agrupamento-FM (*Factorization Machines*)
Ma et al.	Não	Autores apenas realizam um estudo de contas compartilhadas, sem propor solução para o problema	Canais de TV assistidos	-
Nery et al.	Sim, com a vantagem de poder lidar com dados de entrada aninimizados	Análise da Similaridade entre Sessões e Agrupamento das Sessões em perfis	Heterogêneo	Affinity Propagation

O método FIP-SHA, comparado com os trabalhos encontrados no estado da arte, destaca-se por dois motivos. Em sua maioria, as sessões são obtidas pelo tempo delimitado de 30 minutos, isso não representa um indicativo que o usuário interrompeu a sessão e outra pessoa de uma mesma conta está navegando no momento. FIP-SHA analisa as sessões em pares e realiza a interrumpção quando houver um comportamento diferente do esperado. O segundo motivo está presente nos trabalhos apresentarem, na sua maioria, um único escopo homogêneo, e a utilização de um vasto número de metadados, representando uma limitação que impede a utilização dos métodos em sistemas com poucos metadados.

4 FIP-SHA - UM MÉTODO PARA DESCOBERTA DE PERFIS INDIVIDUAIS ATRAVÉS DE CONTAS COMPARTILHADAS

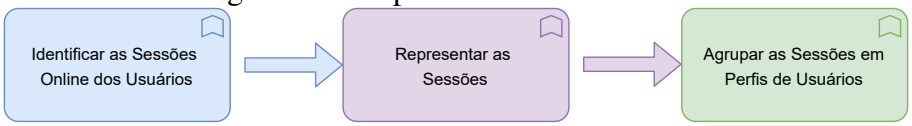
Este Capítulo descreve o FIP-SHA que é um método para descoberta de perfis individuais através de contas compartilhadas. FIP-SHA é um acrônimo para *Finding Individual Profiles through SHared Accounts*. O objetivo principal do FIP-SHA é identificar perfis em contas compartilhadas, através da quebra das sessões individuais presentes em cada conta e agrupar essas sessões, resultando na representação dos perfis dos usuários. FIP-SHA possui como entrada um conjunto de dados representado pelas sessões e os itens visitados (por exemplo, músicas escutadas, filmes assistidos, notícias lidas) e tem por objetivo identificar um ou mais perfis que estão presentes em uma conta compartilhada. A suposição é que as sessões online de um mesmo usuário são semelhantes (por exemplo, o conjunto de itens visitados ou categorias de itens) e, portanto, podem ser agrupadas em perfis de usuários.

Este capítulo inicia apresentando uma visão geral do FIP-SHA e, em seguida, especifica cada etapa do método, a saber: identificação das sessões; representação das sessões; agrupamento das sessões em perfis de usuários. O capítulo é finalizado com um exemplo passo a passo do método proposto.

4.1 Visão Geral

Esta Seção descreve a visão geral do método FIP-SHA que identifica perfis individuais em contas compartilhadas. A principal ideia do trabalho é a construção de perfis de usuários baseados na similaridade entre as sessões dos usuários. A Figura 4.1 ilustra as principais etapas do método proposto, são elas: (i) identificar as sessões *online* dos usuários, através da captura das ações de um usuário em um determinado instante e determinar quando uma sessão começa e termina, representando assim uma nova sessão *online* do usuário; (ii) representar as sessões através dos termos e suas frequências que descrevem os itens visitados; e (iii) agrupar as sessões em perfis de usuários.

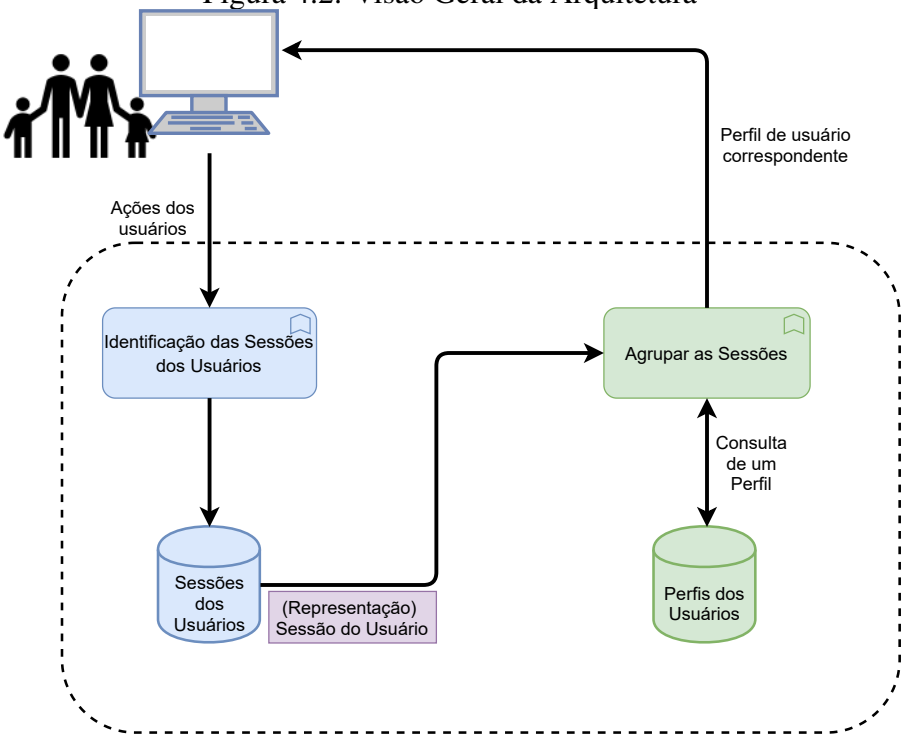
Figura 4.1: Etapas do Método FIP-SHA



A Figura 4.2 ilustra o fluxo da arquitetura proposta para o método FIP-SHA. A

entrada do método são as sessões do usuário, que contém as ações e os itens de interesse, geralmente representados em um sistema real por *logs* de acesso à *Web*, e fornecem informações sobre o comportamento de navegação de um usuário. Essa informação é a entrada para a etapa de identificação das sessões dos usuários e a saída é a representação das sessões geradas através de um conjunto de sessões presentes em cada conta. Na etapa seguinte, o histórico da conta é analisado e as sessões são agrupadas, no qual cada agrupamento representa um perfil de usuário, retornado para a plataforma. Uma evolução do método FIP-SHA, e implementação futura, é considerar a etapa *Online* onde, o perfil é consultado em um banco de dados após realizada a análise da similaridade da sessão corrente com as sessões previamente salvas no sistema, correspondentes aos perfis já analisados a partir das sessões passadas.

Figura 4.2: Visão Geral da Arquitetura



As próximas seções especificam, em detalhes, cada etapa do método FIP-SHA.

4.2 Identificação das Sessões *Online* dos Usuários

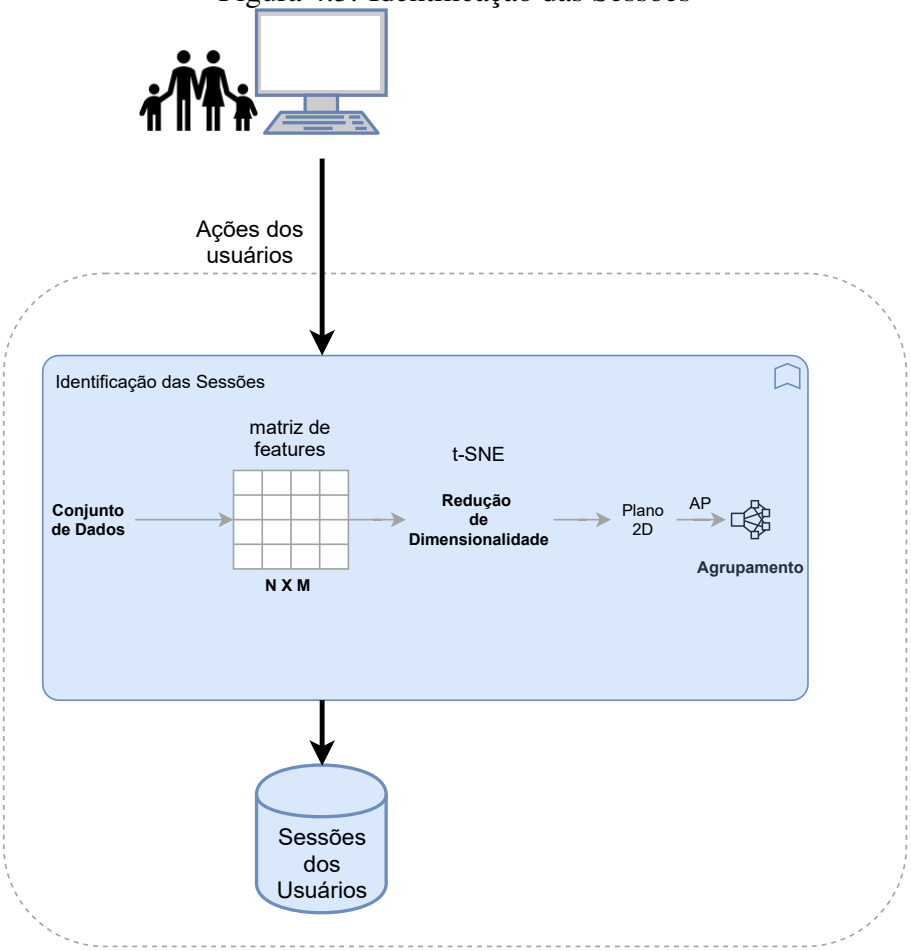
O objetivo da identificação das sessões *online* dos usuários é analisar os itens presentes em uma sessão e assim encontrar um padrão que possa indicar quando existe mais de um usuário na mesma sessão. Um caso trivial é um longo período de inatividade. Por exemplo, no contexto de música, um caso mais complexo, no entanto, é quando um

usuário que estava procurando por músicas *Country* dá lugar a outro que agora ouve *Heavy Metal*. A ideia, por conseguinte, é quando houver um item muito distinto do item em questão, realiza-se uma interrupção e a sessão corrente é dividida, representando assim uma nova sessão *online* do usuário.

O método FIP-SHA identifica as sessões de diferentes perfis de usuário fazendo uma análise por conteúdo, ao invés de quebra de sessões por tempo (*timeout*) diferente da maioria das soluções da literatura atual. A solução adotada para a identificação das sessões foi o uso de técnicas de redução de dimensionalidade e agrupamento de conteúdos para classificar os dados dos usuários em sessões que correspondem ao respectivo perfil do usuário. A redução de dimensionalidade foi escolhida porque tem-se acesso somente as *features* e, quando há conjuntos com muitos dados, enfrenta-se o problema da dimensionalidade. Nos conjuntos de dados utilizados para a validação do método FIP-SHA, tem-se uma grande quantidade de dados e, além disso, o arquivo que contém as *Word Embeddings* possui altas dimensões. A base de dados da Globo (utilizada nos experimentos), por exemplo, possui um arquivo que contém uma matriz com 250 dimensões de *features* e 364.047 artigos de notícias. A técnica de redução de dimensionalidade é necessária para reduzir a complexidade dos dados e aplicar o algoritmo de agrupamento *Affinity Propagation*. A escolha de usar a técnica de agrupamento se justifica por ser uma maneira intuitiva de encontrar grupos em um plano 2D, esse plano é o que melhor representa o conjunto de características que produzem maior relevância ao conjunto de dados.

A Figura 4.3 ilustra a etapa de identificação das sessões. A entrada para o processo de indentificação de sessões é um conjunto de dados, representado por um arquivo de *logs*, onde estão presentes informações do identificador do usuário (*user_id*) para saber qual conta realizou a ação, o *timestamp* para fins de ordenação, o identificador da sessão (*session_id*), e o identificador do conteúdo (*user_id*). A partir dos *logs*, é gerada a Matriz de *Features*. Na matriz está presente o conteúdo representado por $N_{topico} \times M_{dimensoes}$. Em um cenário de notícias, a matriz seria representada por $N_{noticias} \times M_{dimensoes}$. Para saber o que é relevante em uma matriz com essas dimensões, definiu-se o uso da técnica de redução de dimensionalidade t-SNE que calcula através das dimensões existentes e compara em pares o desvio padrão. Um desvio padrão alto significa dizer que determinado conteúdo tem importância, de modo que é obtido o plano que melhor representa estes conteúdos. Na redução de dimensionalidade, o t-SNE modela cada objeto de alta dimensão por um ponto bidimensional de modo que objetos semelhantes sejam modelados por pontos próximos e objetos diferentes sejam modelados por pontos distantes com

Figura 4.3: Identificação das Sessões



alta probabilidade. Desse modo, cada item no conjunto de dados recebe uma coordenada correspondente ao espaço gerado.

A entrada para o t-SNE é o arquivo de *Word Embeddings*. Para a geração desse arquivo, foram utilizadas as bibliotecas *TfidfVectorizer*[1] para construir a matriz esparsa com os elementos armazenados, e a biblioteca NMF[2] (*Non-Negative Matrix Factorization*) para fazer a transformação dos dados. A primeira linha do código do Algoritmo 1 importa o *TfidfVectorizer* do módulo *sklearn.feature_extraction.text*. A terceira linha inicializa o objeto *TfidfVectorizer* chamado *tfidf*, enquanto a quarta linha ajusta e transforma a entrada de dados *data*.

O resultado da execução do tfidf é uma matriz esparsa contendo o número de observações e o número de características. No exemplo do conjunto de dados do Last.fm, a matriz gerada contém dimensões 100000x12107 e a saída apresentada no console foi: "*<100000x12107 sparse matrix of type '<class 'numpy.float64'>' with 6401034 stored elements in Compressed Sparse Row format>*". Após a geração da matriz esparsa, utiliza-

[1]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[2]https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

Algorithm 1: Word Embeddings

Input: data
Output: word embeddings file

```
1 from sklearn.feature_extraction.text import TfidfVectorizer
2 from sklearn.decomposition import NMF
3 tfidf = TfidfVectorizer(max_df=0.6)
4 dat_tfIdf = tfidf.fit_transform(data)
5 nmf = NMF(n_components=6, random_state=0, init='nndsvd')
6 word_embeddings = nmf.fit_transform(dat_tfIdf)
```

mos a biblioteca NMF do módulo *sklearn.decomposition* para fazer a sua decomposição (CICHOCKI; PHAN, 2009). O método *fit_transform* recebe como parâmetro a matriz de dados a ser decomposta e retorna os dados transformados.

Após gerar o plano 2D, é preciso encontrar os centróides dos agrupamentos dos itens visitados ou categorias (no cenário de notícias), para isso foi utilizado o método *Affinity Propagation*. Dessa maneira, consegue-se fazer uma análise tentando modificar a granularidade (muitos tipos de tópicos versus poucos tipos de tópicos) e consegue-se encontrar as suas coordenadas. FIP-SHA utiliza o *Affinity Propagation*, baseado no trabalho de Moreira (MOREIRA; FERREIRA; CUNHA, 2018), que também adotou o *Affinity Propagation* e t-SNE. O critério utilizado para encontrar qual agrupamento cada item pertence foi o centróide que obteve a menor distância euclidiana. O método escolhido para encontrar os limites da sessão baseou-se na vontade de que os usuários tendem a escolher itens pertencentes aos mesmos tópicos ou semelhantes. Esta hipótese foi usada para criar o modelo usando a distância euclidiana dos tópicos para medir quão diferentes eles são. Portanto, o centro de cada agrupamento do tópico é usado para calcular a diferença semântica de cada tópico com base na distância euclidiana do par de tópicos sendo comparados. Dessa forma, consegue-se inferir se o conteúdo analisado pertence ao mesmo tópico ou não, se estiverem agrupados próximos significa que podem ser representados pelo mesmo. A quebra de sessão é realizada ao se analisar a frequência dos tipos de conteúdos e se observar um conteúdo diferente do perfil. Algoritmo 2 fornece uma visão geral da estratégia de corte nas sessões.

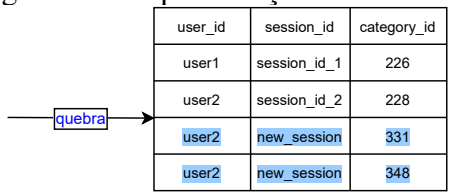
Algorithm 2: FIP-SHA - Split Sessions

```
   Input: L = < u,s,i >
   Output: split of sessions into shared accounts
1  constant CUTOFF_THRESHOLD
2  read features
3  generate tsne file
4  Function get_cluster_centers:
5      clustering = AffinityPropagation.fit_transform
6      cluster_centers_indices = clustering.cluster_centers_indices_
7      labels = clustering.labels_
8      return cluster_centers, labels
9  get random labels
10 get corresponding centroid labels
11 map euclidean from centroid
12 Function cut_sessions_based_on_cutoff_threshold:
13     for session ∈ sessions do
14         if distance > CUTOFF_THRESHOLD then
15             cut current session into a new one;
16         end
17     end
18     return sessions
```

4.3 Representação das Sessões

As sessões são representadas através dos termos e suas frequências que descrevem os itens visitados. A entrada para essa etapa são as sessões após realizados os devidos cortes, e a saída são as sessões e os termos relevantes prontos para a etapa seguinte de agrupamento de sessões.

Figura 4.4: Representação das Sessões



A Figura 4.4 ilustra uma conta com o devido corte na sessão, mantendo para a próxima etapa somente o conteúdo relevante. Sendo elas, as informações de *session_id* e id da categoria/item, informações essas que descrevem a frequência do item analisado.

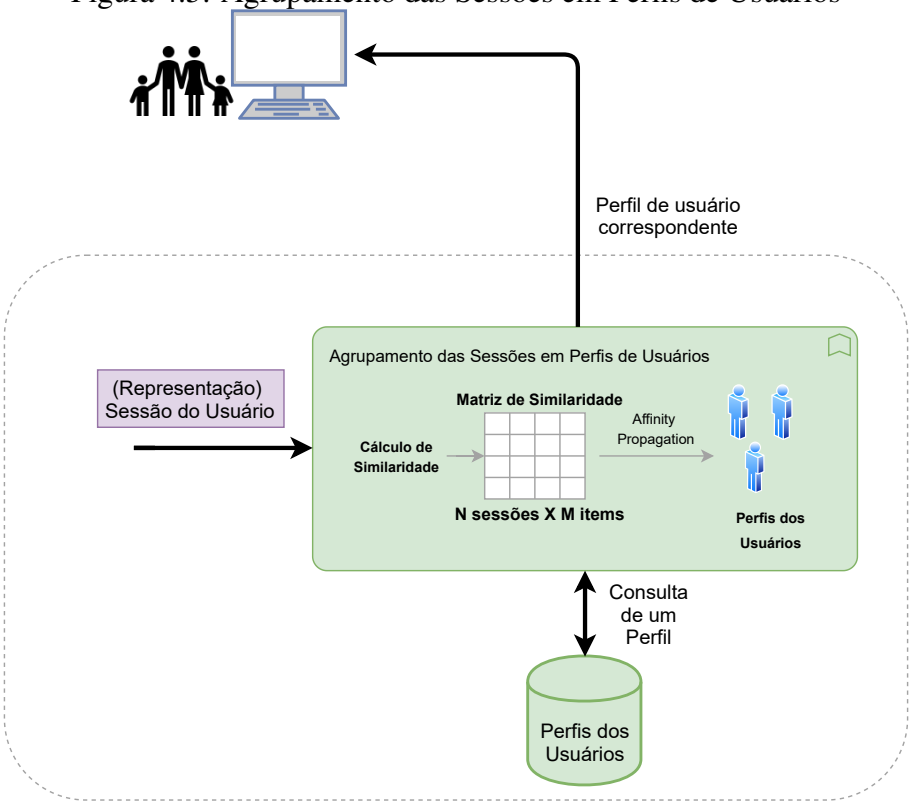
4.4 Agrupamento das Sessões em Perfis de Usuários

As etapas para a realização do agrupamento das sessões e a geração dos perfis dos usuários são descritas nesta seção. A entrada para esta etapa são as sessões após o corte, quando necessário, representado pela primeira etapa do método FIP-SHA, e a saída são os perfis identificados e presentes em cada conta. A ideia é, analisando a similaridade entre as sessões, agrupar e identificar as sessões e, dessa forma, os perfis dos usuários presentes

em cada conta compartilhada.

Para o agrupamento das sessões em perfis de usuários foi utilizado o método de agrupamento chamado *Affinity Propagation*, referida no capítulo 2.5.3. Aqui, somente o uso dentro da arquitetura do FIP-SHA é exemplificado, conforme ilustrado na Figura 4.5. O método *Affinity Propagation* encontra o número de agrupamentos automaticamente, sendo estes representados pelo elemento que melhor generaliza todos os elementos dentro do agrupamento. Como o número de usuários em uma conta é desconhecido, o *Affinity Propagation* tornou-se um bom candidato. A entrada para a etapa de agrupamento são as sessões processadas pelas etapas de Identificação 4.2 e Representação de Sessões 4.3. O agrupamento é dividido em quatro etapas: (i) o cálculo da similaridade, onde as sessões de cada conta são comparadas em pares utilizando a métrica de similaridade por Cosseno, (ii) e o resultado é a Matriz de Similaridade entre as sessões em cada conta. Após, a similaridade entre as sessões é utilizada no algoritmo *Affinity Propagation* para realizar o (iii) agrupamento das sessões e (iv) gerar os grupos que representam os perfis dos usuários.

Figura 4.5: Agrupamento das Sessões em Perfis de Usuários



O objetivo da similaridade é calcular a similaridade entre as sessões, considerando os itens de cada sessão e quantas vezes os itens estão presentes na mesma. Para calcular a similaridade entre os itens, testamos duas métricas, Cosseno e Jaccard, porém a similaridade por Cosseno foi a que apresentou o melhor resultado entre as duas.

FIP-SHA trabalha com a identificação de usuários em sessões, analisando todo o histórido da conta (denominado neste trabalho de sessões passadas), contendo a sequência de sessões de uma conta S_a. Essa etapa consiste em agrupar cada sessão $s \in S$ em k agrupamentos que representam os usuários, $C_a = \{c_1^a, c_2^a, ..., c_{Ka}^a\}$ de forma que as sessões de um mesmo usuário sejam representadas no mesmo agrupamento. A métrica de similaridade entre as sessões é baseada nos itens $i \in I$ compartilhados entre elas.

O Algoritmo 3 apresenta a estratégia para agrupar as sessões em perfis de usuários. Primeiro, o cálculo da similaridade, no qual os pares das sessões da conta são comparadas utilizando a métrica de similaridade por Cosseno, é apresentado. Em seguida, a utilização do *Affinity Propagation* no processo de agrupamento é descrita. Por fim, é gerada uma estrutura de dados que representa os perfis obtidos.

Algorithm 3: FIP-SHA - Cluster Sessions

```
   Input: split of sessions into shared accounts
   Output: profiles
1  for account ∈ dataset do
2      fetch list of sessions as items lists;
3      obtain item matrix from each session;
4      similarities = similarity(session_item_lists)
5      clustering = AffinityPropagation.fit(similarities)
6  end
7  return clusters;
8
```

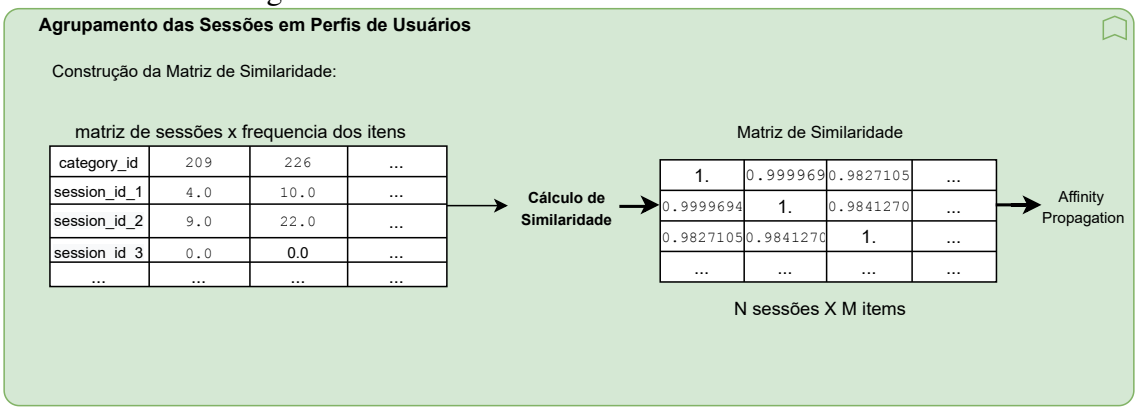
A seguir, as etapas que fazem parte do agrupamento são apresentadas.

4.4.1 Cálculo da Similaridade das sessões por Cosseno

O objetivo é realizar o cálculo da similaridade entre as sessões, comparando-as par a par. O resultado deste cálculo é a entrada para o método *Affinity Propagation*, que necessita da matriz de semelhanças dos dados como entrada. Foi escolhida a métrica de similaridade por Cosseno, citação que se encontra referida no capítulo 2.2.1, para realizar o cálculo. As sessões são representadas por vetores, no qual cada espaço do vetor corresponde a um item que o usuário interagiu e o valor é a quantidade de vezes que a interação ocorreu com esse item. A Figura 4.6 exemplifica os passos descritos acima.

A métrica do Cosseno utiliza os itens em comum entre as sessões para calular a similaridade, e é necessário que esses itens se repitam frequentemente entre as sessões. Para os itens que o usuário interagiu, o número de interações com cada um deles é a média próxima de 1. Mas por outro lado em um site de notícias, por exemplo, são raros os casos em que um usuário lê uma mesma notícia repetidas vezes, isso implica que

Figura 4.6: Matriz de Similaridade entre as Sessões



é frequente a comparação entre duas sessões resultar em zero, portanto, nesses casos utilizamos a categoria das notícias para o cálculo da similaridade, resultando em um valor mais preciso.

4.4.2 Agrupamento por Affinity Propagation

O algoritmo de agrupamento utilizado para formar os perfis de usuários, agrupando as sessões por similaridade, é descrito nesta seção. O *Affinity Propagation* é um algoritmo de agrupamento, ao contrário dos algoritmos de agrupamento *k-means* ou *k-medoids*, não requer a estimativa do número de grupos antes de executar o algoritmo. Similar ao *k-medoids*, o algoritmo encontra ”exemplares”, ou seja, membros do conjunto de entrada que são representações do grupo.

A entrada são as semelhanças entre os pontos de dados e o algoritmo identifica exemplares com base em certos critérios. As mensagens são trocadas entre os pontos de dados até que um conjunto de exemplares de alta qualidade seja obtido. Para isso, precisamos calcular as matrizes de similaridade, responsabilidade (*responsibility*) e disponibilidade (*availability*).

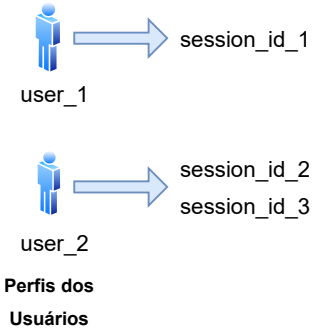
A matriz de similaridade contem informações sobre a similaridade entre quaisquer instâncias, neste trabalho, entre duas sessões $s(i, k)$, calculada através da similaridade por Cosseno. A similaridade de um item com ele mesmo $s(k, k)$ é chamada de preferência, e os maiores valores de preferência torna o item mais provável de ser escolhido como centro do agrupamento. A responsabilidade $r(i, k)$ quantifica quão adequado é o elemento k, para ser um exemplo para o elemento i. Já a disponibilidade $a(i, k)$ quantifica o quão apropriado é para i escolher k como seu exemplar, levando em consideração a preferência de outros pontos por k como um exemplar. As matrizes R e A são atualizadas iterativa-

mente. Este procedimento pode ser encerrado após um número fixo de iterações, após alterações nos valores obtidos ficarem abaixo de um limite, ou após os valores permanecerem constantes por algum número de iterações.

Somando as matrizes de Responsabilidade (R) e Disponibilidade (A) obtem-se as informações de agrupamento necessárias. Esse calculo é realizado após o término da atualização. Um elemento i será atribuído a um exemplar k que não é apenas altamente responsável, mas também altamente disponível para i. O elemento com o valor mais alto em cada linha é designado para ser um exemplar e os elementos correspondentes que compartilham o mesmo exemplar são agrupados.

O cálulo de evidências é a principal tarefa do algoritmo de *Affinity Propagation*. Primeiro, os parâmetros número máximo de iterações, fator de amortecimento (*damping*), que influenciam no desempenho do agrupamento, e o valor de ”preferência” são definidos. A calibragem dos parâmetros, isto é, ajustes e testes, é demonstrada no Capítulo 5 de exeprimentos. As matrizes são inicializadas com zero para a primeira iteração e, em seguida, o procedimento de passagem por mensagens é considerado e as responsabilidades e as disponibilidades são atualizadas a partir da matriz de similaridade. Os exemplos são então decididos com base na semelhança máxima para encontrar aglomerados. Finalmente, os exemplos são definidos em um agrupamento e o processo é concluído. Obtendo como resultado os perfis de usuários, Figura 4.7, representados pelos agrupamentos encontrados, que contém as sessões agrupadas pelas suas similaridades.

Figura 4.7: Representação dos Perfis dos Usuários



4.5 Exemplo de Execução do FIP-SHA

Nesta seção, um exemplo de execução do FIP-SHA é apresentado, sendo as principais etapas: parametrização dos algoritmos t-SNE e *Affinity Propagation*, geração do arquivo de *Word Embeddings* a partir do conjunto de *logs* importado, quebra das sessões

após atingir um determinado valor, análise da similaridade entre as sessões, e por fim, o agrupamento das sessões em perfis de usuários. O exemplo utiliza a base de dados da Globo, que é a mesma utilizada posteriormente nos experimentos.

A primeira etapa, antes de iniciar a execução, é a preparação dos algoritmos t-SNE e *Affinity Propagation* através da parametrização. Para o algoritmo t-SNE, a escolha do parâmetro *perplexity* é fundamental para o seu funcionamento correto, e por isso, foram testados diferentes valores para o mesmo (5, 2, 30 e 50). No algoritmo *Affinity Propagation*, o parâmetro *damping* precisou ser definido também, para isso foram testados os valores 0.5, 0.6, 0.7, 0.8 e 0.9. A calibragem dos parâmetros, isto é, ajustes e testes, é demonstrada no Capítulo 5 de experimentos.

Após a configuração adequada, inicia-se a execução do algoritmo importando os arquivos de *logs* que contém as interações dos usuários nas sessões, Tabela 4.1. Por exemplo, o usuário com id 0 leu o artigo com id 157541 que pertence à categoria com id 281 e essa interação ocorreu na sessão que possui o id 1506825423271737.

Tabela 4.1: Arquivo de interações dos usuários

	user_id	session_id	session_start	session_size	click_article_id	click_timestamp	click_referrer_type	article_id	category_id
0	0	1506825423271737	1506825423000	2	157541	1506826828020	2	157541	281
1	20	1506825727279757	1506825727000	2	157541	1506836548634	1	157541	281

O arquivo de *Word Embeddings* é gerado através dos *logs* de entrada, Figura 4.8, e o resultado é a entrada para o algoritmo t-SNE. Com o resultado obtido, encontra-se os centros dos agrupamentos através do algoritmo *Affinity Propagation*.

Figura 4.8: Arquivo de Embeddings da Globo

```
array([[-0.16118301, -0.95723313, -0.13794445, ..., -0.231686  ,
         0.5974159 ,  0.40962312],
       [-0.52321565, -0.974058  ,  0.73860806, ...,  0.18282819,
         0.39708954, -0.83436364],
       [-0.61961854, -0.9729604 , -0.20736018, ..., -0.44758022,
         0.8059317 , -0.28528407],
       ...,
       [-0.25139043, -0.9762427 ,  0.58609664, ..., -0.14372464,
         0.06809307, -0.7050104 ],
       [ 0.22434181, -0.92328775, -0.38174152, ...,  0.6871319 ,
        -0.5315117 ,  0.01072566],
       [-0.25713393, -0.9946313 ,  0.9837918 , ...,  0.98387307,
        -0.8381829 , -0.1792827 ]], dtype=float32)
```

Após encontrar os centros dos agrupamentos com o algoritmo *Affinity Propagation*, constroi-se as *labels* fictícias de cada ponto, valores representados pelas coordenadas X e Y, e assim consegue-se associar à respectiva coluna (*article_id*), Tabela 4.2.

Com as coordenadas X e Y, calcula-se a distância euclidiana de cada sessão. A ideia é, assim que a distância atingir um determinado delimitador (nomeamos de "*cut*

Tabela 4.2: Coordenadas X e Y de cada article_id, Globo

	article_id	X	Y
0	3	34.92	0.09
1	27	10.95	-26.21
2	69	34.35	-0.17
3	81	34.41	-0.69
4	84	35.33	-1.57
...	...	...	...
46028	364017	-8.20	2.88
46029	364022	-16.06	-1.16
46030	364028	-0.11	14.09
46031	364043	4.73	-8.28
46032	364046	4.73	-8.27

off") é realizada a quebra da sessão corrente em uma nova sessão. No exemplo da Tabela 4.3, realiza-se o corte na sessão ”1506875407922788”, que atingiu a distância de aproximadamente 41.60, acima do ponto de corte definido pelo experimento (*cut off* = 40) e assim, encerra-se a primeira etapa do *Running Example 1*.

Tabela 4.3: Corte em uma determinada sessão, Globo

	user_id	session_id	click_article_id	x_centroid	y_centroid	distance
0	15275	1506875407922788	101192	-23.93	-19.91	0.00
1	15275	1506875407922788	102738	4.27	-4.04	32.36
2	15275	1506875407922788	102701	4.27	-4.04	0.00
3	15275	1506875407922788	102692	16.82	-11.57	14.63
4	15275	1506875407922788_000	101193	-23.93	-19.91	**41.59**
5	15275	1506875407922788_000	102696	4.27	-4.04	32.36
6	15275	1506875407922788_000	102661	0.05	1.41	6.90
...	...	...	...	...	...	...
14	15275	1506889585128598	101195	-23.93	-19.91	32.36

A saída do *Running Example 1*, e entrada do *Running Example 2*, são as sessões após realizado algum corte. Existem duas etapas, são elas: o cálculo de similaridade, onde as sessões de cada conta são comparadas em pares usando a métrica de similaridade por Cosseno, resultando na Matriz de Similaridade entre as sessões em cada conta. Posteriormente, a similaridade entre as sessões é utilizada como entrada para o algoritmo *Affinity Propagation* para realizar o agrupamento das sessões e gerar os grupos que melhor representam os perfis dos usuários.

A frequência dos itens é representada através da construção da Matriz de Sessões, 4.4. As sessões são representadas por vetores, no qual cada espaço do vetor corresponde a um item que a sessão interagiu e o valor é a quantidade de vezes que a interação ocorreu com esse item.

A métrica do Cosseno utiliza os itens em comum entre as sessões para calular a similaridade e construir a Matriz de Similaridade, Figura 4.9. Para os itens que tiveram maior interação, o número de interações é a média próxima ou igual a 1.

A última etapa, após o cálculo da similaridade, é encontrar os agrupamentos que representam os perfis dos usuários nas contas. A Tabela 4.5 representa o resultado da

Tabela 4.4: Matriz de Sessões, Globo

category_id session_id	147	209	226	228	247	250	254	281	289	317
1506875407922788	0.0	0.0	4.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0
1506889585128598	0.0	0.0	9.0	22.0	0.0	0.0	0.0	0.0	0.0	0.0
1506901762871153	0.0	0.0	14.0	22.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
1507562453117515	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0

Figura 4.9: Matriz de Similaridade

```
array([[1.        , 0.99996948, 0.98271058, ..., 0.99215722, 1.        ,
        0.99949648],
       [0.99996948, 1.        , 0.98412702, ..., 0.99115044, 0.99996948,
        0.99921809],
       [0.98271058, 0.98412702, 1.        , ..., 0.95186056, 0.98271058,
        0.97634099],
       ...,
       [0.99215722, 0.99115044, 0.95186056, ..., 1.        , 0.99215722,
        0.99562378],
       [1.        , 0.99996948, 0.98271058, ..., 0.99215722, 1.        ,
        0.99949648],
       [0.99949648, 0.99921809, 0.97634099, ..., 0.99562378, 0.99949648,
        1.        ]])
```

conta que contém 2 usuários e possui id de conta 0.0. A saída do algoritmo apresenta como resultado o número de agrupamentos gerados (no exemplo, são 11) e, as *labels* de cada ponto [4, 4, 4, 4, 4, 0, 4, 4, 7, 1, 4, 4, 4, 4, 2, 4, 3, 4, 4, 5, 6, 7, 8, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 9, 4, 10, 4, 4, 4, 4, 4, 4, 4], as *labels* são atribuídas às sessões e o resultado são os perfis dos usuários que contém as sessões.

Tabela 4.5: Labels e Sessões, Globo

session_id	label	usuário
1506875407922788	4	15275
1506889585128598	4	15275
1506901762871153	4	15275
1506973853411261	4	15275
1506996930213065	4	15275
1507031176212612	0	123929
1507048381281755	4	15275
1507081227272713	4	15275
1507229034752109	7	15275
...	...	...
1508123293257891	4	15275

5 EXPERIMENTOS

Este capítulo descreve os experimentos realizados para avalidar as etapas do método FIP-SHA proposto nesta dissertação para a descoberta de perfis individuais através de Contas Compartilhadas. São relizados quatro experimentos:

- Experimento #1: avaliação do desempenho do corte em sessões;
- Experimento #2: parametrização do algoritmo t-SNE;
- Experimento #3: calibragem do parâmetro *Damping* e desempenho do algoritmo *Affinity Propagation*;
- Experimento #4: avaliação dos agrupamentos.

Este capítulo começa descrevendo as bases de dados reais e a construção das bases de dados que sinteticamente simulam as conta compartilhadas. Em seguida, são descritas as ferremtentas utilizadas e, cada experimento é descrito em detalhes, acompanhado pela discussão dos resultados obtidos.

O código-fonte do projeto, juntamente com os *scripts* de avaliação, os dados usados para realizar os experimentos e os resultados obtidos, estão disponíveis publicamente no GitLab[1]. Para o desenvolvimento do projeto foi utilizada a linguagem de programação *Python* e as bibliotecas complementares *sklearn*, *pandas* e *numpy*.

5.1 Bases de dados

Para realização dos experimentos, foram utilizadas duas bases de dados. A primeira, Globo[2], armazena *logs* de interações de usuários do portal de notícias G1. A segunda, Lastfm[3], armazena sessões de escuta de músicas da plataforma Lastfm.

Cabe ressaltar que não foram encontradas bases de dados com informações sobre contas compartilhadas disponibilizadas publicamente. Neste trabalho, as bases de dados são sinteticamente unificadas para representar a ideia de contas compartilhadas através da união de dois ou mais identificadores. A Tabela 5.1 mostra detalhes da estrutura das bases de dados coletadas, como período, quantidade de sessões e itens visitados e o número de usuários. Nas próximas subseções, cada base de dados é descrita em detalhes bem como a construção das contas compartilhadas para fins de avaliação nos experimentos.

[1] https://gitlab.com/carolinaNery94/accountprofiles-mscproject
[2] <https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom>
[3] <http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html>

Tabela 5.1: Informações das bases de dados utilizados.

	Período	Sessão/Itens visitados	Número de usuários
Globo	Outubro 1 a 16, 2017	1.048.594	322.897
Lastfm	(Publicado em Maio, 2010)	907.887	992

5.1.1 Globo

A base de dados da Globo[4] é um conjunto de arquivos que representa os *logs* de interações dos usuários (visualizações de páginas) do portal de notícias do G1 criada e utilizada nos trabalhos desenvolvidos por (MOREIRA; FERREIRA; CUNHA, 2018) e (MOREIRA; JANNACH; CUNHA, 2019) e posteriormente disponibilizada na plataforma *Kaggle*. A base de dados contém dados de 1 a 16 de outubro do ano de 2017, incluindo cerca de 3 milhões de cliques, distribuídos em mais de 1 milhão de sessões de 320.000 usuários que leram mais de 46.000 notícias diferentes durante esse período.

Tabela 5.2: Dicionário da base de dados da Globo

Coluna	Descrição
user_id	id do usuário
session_id	id da sessão
session_start	primeira interação da sessão
session_size	número de interações da sessão
click_article_id	id do artigo interagido pelo usuário
click_amp	timestamp da interação
click_environment	id do ambiente
click_deviceGroup	id do tipo de dispositivo
click_os	id do sistema operacional
click_country	id do pais
click_region	id da região
click_referrer_type	tipo do clique
article_id	identificador do artigo
category_id	id da categoria do artigo
created_at_ts	data de publicação do artigo
publisher_id	id do editor do artigo
words_count	número de palavras no artigo

A Tabela 5.2 apresenta os atributos da base de dados, as principais informações presentes são os artigos lidos por usuário, a categoria e outras informações referentes a cada artigo, levando-se em consideração um período de tempo. Os atributos utilizados nos experimentos estão em negrito. As principais características dessa base de dados são:

1. Sessões são os intervalos de atividade do usuário no site. O "corte", ou quebra da sessão, é de um intervalo a cada 30 minutos decorridos;
2. Muitas sessões são curtas (60% das sessões com apenas 2 itens e 21% com 3);
3. Usuários raramente clicam na mesma notícia mais de uma vez, nem na mesma sessão nem em outras.

[4]<https://www.kaggle.com/gspmoreira/news-portal-user-interactions-by-globocom>

Considerando a característica do item 3, as sessões de um mesmo usuário apresentam similaridades baixas porque raramente contém itens em comum. Nesse caso, optou-se por utilizar a categoria (*category_id*) das notícias como item de agrupamento, informação que se repete mais comumente no conjunto de dados. A Figura 5.1 contém uma amostragem do arquivo utilizado.

Figura 5.1: Exemplo entrada de dados Globo

	user_id	session_id	session_start	session_size	click_article_id	click_timestamp	click_referrer_type	article_id	category_id
0	0	1506825423271737	1506825423000	2	157541	1506826828020	2	157541	281
1	20	1506825727279757	1506825727000	2	157541	1506836548634	1	157541	281
2	44	1506826139185781	1506826139000	5	157541	1506857278141	1	157541	281
3	45	1506826142324782	1506826142000	2	157541	1506827309970	1	157541	281
4	76	1506826463226813	1506826463000	2	157541	1506828823469	1	157541	281
...	...	...	...	...	...	...	...	...	...
2988176	195186	1508210422411129	1508210422000	4	2221	1508210469562	1	2221	1
2988177	75658	1508210696185183	1508210696000	4	271117	1508210951703	2	271117	399
2988178	217129	1508210976336246	1508210976000	2	20204	1508210990810	5	20204	9
2988179	217129	1508210976336246	1508210976000	2	70196	1508211020810	5	70196	136
2988180	51099	1508211320193320	1508211320000	2	98243	1508211782523	5	98243	220

5.1.1.1 Conta Compartilhada

A ideia de conta compartilhada é modelada a partir dos dados de cliques de usuários. Todos os cliques dos usuários selecionados foram ordenados por *timestamp* e, então, agrupados em uma única conta. Nas Tabelas 5.4, 5.5, 5.6, dois, três e quatro usuários estão presentes, respectivamente, e simulam as contas compartilhadas com as sessões dos usuários que foram agrupadas. A Tabela 5.3 apresenta em detalhes a relação do número de usuários por conta e a quantidade de contas simuladas.

Tabela 5.3: Resumo da base de dado da Globo usado na nossa avaliação.

Número de usuários	Qtd. Contas Simuladas	10%
2	16.645	1.702 contas
3	11.097	1.134 contas
4	8.322	851 contas

Para os experimentos, foi utilizado 10% do total de contas simuladas para podermos analisar os resultados obtidos em maiores detalhes. No exemplo a seguir, são utilizadas as contas que possuem id de conta (*account_id*) 0.0, esse valor é incremental e foi inserido para simular a conta compartilhada. Para as contas que possuem 2 usuários, foi realizada a união das sessões do *usuário1* com as sessões do *usuário2*. Já a conta que possui 3 usuários é a união das sessões do *usuário1* com as sessões do *usuário2*, acrescida das sessões do *usuário3*. A mesma lógica foi aplicada para uma conta que possui 4 usuários.

Tabela 5.4: Conta com 2 usuários (15275 e 123929)

	user_id	session_id	category_id	account_id
0	15275	1506875407922788	226	0.0
1	15275	1506875407922788	226	0.0
2	15275	1506875407922788	226	0.0
3	15275	1506875407922788	226	0.0
4	15275	1506875407922788	228	0.0
...	...	...	...	...
770	123929	1507548333183379	437	0.0
771	123929	1507548333183379	437	0.0
772	123929	1507906094135594	437	0.0
773	123929	1507906094135594	147	0.0
774	123929	1507906094135594	435	0.0

Tabela 5.5: Conta com 3 usuários (15275, 123929 e 9913)

	user_id	session_id	category_id	account_id
0	15275	1506875407922788	226	0.0
1	15275	1506875407922788	226	0.0
2	15275	1506875407922788	226	0.0
3	15275	1506875407922788	226	0.0
4	15275	1506875407922788	228	0.0
...	...	...	...	...
825	9913	1507293100239408	327	0.0
826	9913	1507059984303172	327	0.0
827	9913	1507059984303172	327	0.0
828	9913	1507576825158393	327	0.0
829	9913	1507576825158393	26	0.0

Tabela 5.6: Conta com 4 usuários (15275, 123929, 9913 e 84120)

	user_id	session_id	category_id	account_id
0	15275	1506875407922788	226	0.0
1	15275	1506875407922788	226	0.0
2	15275	1506875407922788	226	0.0
3	15275	1506875407922788	226	0.0
4	15275	1506875407922788	228	0.0
...	...	...	...	...
862	84120	1507116757158039	331	0.0
863	84120	1507116757158039	348	0.0
864	84120	1507116757158039	174	0.0
865	84120	1507739570144688	331	0.0
866	84120	1507739570144688	348	0.0

5.1.2 Lastfm

O conjunto de dados do Lastfm contém tuplas `<usuário, registro de data e horário (timestamp), artista, música>` coletadas da API do Lastfm. Esse conjunto representa todos os hábitos de escuta para 992 usuários durante o ano de 2009. O arquivo utilizado no experimento, contém informações do usuário, artista e música, e possui um total de 19.098.862 linhas, considerado o maior e mais completo conjunto de dados para o estudo em questão. Na Figura 5.2 está um exemplo das primeiras 5 linhas do conjunto.

Analisando a Figura 5.3, o nome das músicas por transação (*track_name*) representa um bom indicador de dados para os experimentos, pois existe uma repetição de valores tanto para os usuários *user_000001* e *user_000002*, um indicador para representar um padrão do usuário logado.

Figura 5.2: Exemplo arquivo 1, lastFM

	user_id	timestamp	artist_id	artist_name	track_id	track_name
0	user_000001	2009-05-04T23:08:57Z	f1b1cf71-bd35-4e99-8624-24a6e15f133a	Deep Dish	NaN	Fuck Me Im Famous (Pacha Ibiza)-09-28-2007
1	user_000001	2009-05-04T13:54:10Z	a7f7df4a-77d8-4f12-8acd-5c60c93f4de8	坂本龍一	NaN	Composition 0919 (Live_2009_4_15)
2	user_000001	2009-05-04T13:52:04Z	a7f7df4a-77d8-4f12-8acd-5c60c93f4de8	坂本龍一	NaN	Mc2 (Live_2009_4_15)
3	user_000001	2009-05-04T13:42:52Z	a7f7df4a-77d8-4f12-8acd-5c60c93f4de8	坂本龍一	NaN	Hibari (Live_2009_4_15)
4	user_000001	2009-05-04T13:42:11Z	a7f7df4a-77d8-4f12-8acd-5c60c93f4de8	坂本龍一	NaN	Mc1 (Live_2009_4_15)

Figura 5.3: *track_name* duplicado para o *user_000001* e *user_000002*, respectivamente

	user_id	track_name
6378	user_000001	'84 Pontiac Dream
6747	user_000001	'84 Pontiac Dream
12910	user_000001	'84 Pontiac Dream
13336	user_000001	'84 Pontiac Dream
13722	user_000001	'84 Pontiac Dream
...	...	...
16318	user_000001	高木正勝 Vs 南博
16325	user_000001	高木正勝 Vs 南博
12453	user_000001	１５の恋
12755	user_000001	１５の恋
12826	user_000001	１５の恋

	user_id	track_name
18257	user_000002	$100 Cover
18687	user_000002	$100 Cover
19087	user_000002	(Feel It) Day By Day
20532	user_000002	(Feel It) Day By Day
20637	user_000002	(Feel It) Day By Day
...	...	...
21148	user_000002	Þversögn
23190	user_000002	Þversögn
29229	user_000002	Þversögn
18250	user_000002	...
18680	user_000002	...

As principais propriedades dessa base de dados são:

1. As sessões são longas (52% têm mais de 10 itens);
2. Usuários não repetem as músicas na mesma sessão (apenas 9% das sessões possuem músicas repetidas);
3. Usuários repetem músicas entre várias sessões (52% das vezes que um usuário ouviu uma música, ele a ouviu novamente em outra sessão).

Cabe ressaltar que na base de dados Globo, a categoria da notícia foi utilizada como forma de agrupamento, pois um usuário raramente lê a mesma notícia mais de uma vez por sessão. Já no Lastfm esse problema não ocorre.

5.1.2.1 Conta Compartilhada

Na base de dados do Lastfm não existe a ideia de conta compartilhada e as músicas escutadas pelos usuários não estão separadas pelo conceito de sessão, diferente da base de dados da Globo, ou seja, não existe um valor de *session_id*. Por esse motivo, foi necessário criar o identificador das sessões (*session_id*), sendo utilizado como base o tempo (*timestamp*). A Tabela 5.7 apresenta em detalhes a relação do número de usuários por conta e a quantidade de contas simuladas.

Após criado o conceito se sessão, pôde-se moldar as contas compartilhadas com

Tabela 5.7: Resumo da base de dado do Lastfm usado na nossa avaliação.

Número de usuários	Qtd. Contas Simuladas	10%
2	490	49 contas
3	327	32 contas
4	245	24 contas

dois, três e quatro usuários, respectivamente, exemplificadas nas Tabelas 5.8, 5.9 e 5.10. A conta que possui 2 usuários é a união das sessões do *usuario1* com as sessões do *usuario2*. Em uma conta que possui 3 usuários, é a união das sessões do *usuario1*, com as sessões do *usuario2* e as sessões do *usuario3*. Já para uma conta que possui 4 usuários, a conta é simulada com a união das sessões dos usuarios *usuario1*, *usuario2*, *usuario3* e *usuario4*.

Tabela 5.8: Conta com 2 usuários (user_000666 e user_000735)

	user_id	session_id	track_name	account_id
0	user_000666	597228	Czarnuch	0.0
1	user_000666	597228	Iv L.O. '98	0.0
2	user_000666	597228	King	0.0
3	user_000666	597496	Czarnuch	0.0
4	user_000666	597496	Kombajn Bizon	0.0
...	...	...	...	...
19131	user_000735	658620	Stolle Nordmenn	0.0
19132	user_000735	658620	Deg Og Meg Anei!	0.0
19133	user_000735	658619	King Of The Mountain Road Master	0.0
19134	user_000735	658656	P320080605-Emsmith	0.0
19135	user_000735	658661	Ode To...	0.0

Tabela 5.9: Conta com 3 usuários (user_000666, user_000735 e user_000338)

	user_id	session_id	track_name	account_id
0	user_000666	597228	Czarnuch	0.0
1	user_000666	597228	Iv L.O. '98	0.0
2	user_000666	597228	King	0.0
3	user_000666	597496	Czarnuch	0.0
4	user_000666	597496	Kombajn Bizon	0.0
...	...	...	...	...
62499	user_000338	322508	Liian Kauan	0.0
62500	user_000338	322508	Mun Koti Ei Oo Täällä	0.0
62501	user_000338	322508	Tämä Rakkaus	0.0
62502	user_000338	322490	Mun Koti Ei Oo Täällä	0.0
62503	user_000338	322519	Football Weekly: Champions League Semi-Final P...	0.0

Tabela 5.10: Conta com 4 usuários (user_000666, user_000735, user_000338 e user_000095)

	user_id	session_id	track_name	account_id
0	user_000666	597228	Czarnuch	0.0
1	user_000666	597228	Iv L.O. '98	0.0
2	user_000666	597228	King	0.0
3	user_000666	597496	Czarnuch	0.0
4	user_000666	597496	Kombajn Bizon	0.0
...	...	...	...	...
108818	user_000338	322508	Liian Kauan	0.0
108819	user_000338	322508	Mun Kot Ei Oo Täällä	0.0
108820	user_000338	322508	Tämä Rakkaus	0.0
108821	user_000338	322490	Mun Kot Ei Oo Täällä	0.0
108822	user_000338	322519	Football Weekly: Champions League Semi-Final P...	0.0

5.2 Experimento 1 - avaliação do desempenho do corte em sessões

O objetivo do experimento é, a partir de um conjunto de sessões em uma conta compartilhada, e dado um ponto de corte, realizar a quebra de uma determinada sessão em duas ou mais sessões.

Com o arquivo de *Word Embeddings*, reduz-se a dimensionalidade usando o t-SNE. Aqui vale mencionar que o t-SNE tem um hiperparâmetro chamado *perplexity*, esse parâmetro equilibra a atenção que o t-SNE dá aos aspectos locais e globais dos dados e pode ter efeito no gráfico resultante, é aproximadamente uma estimativa do número de vizinhos próximos que cada ponto possui. Um conjunto de dados mais denso geralmente requer um valor do parâmetro *perplexity* mais alto.

Um erro que pode ser cometido ao executar o t-SNE é escolher um valor para o parâmetro *perplexity* e não testar os resultados com outros valores. A chave é garantir que o algoritmo seja executado por tempo suficiente para se estabilizar, similar ao algoritmo *Affinity Propagation*. Para isso, foi criado o experimento que ajusta esse valor, seção 5.3, onde diferentes valores junto às bases de dados utilizadas são avaliados.

Com a presença das coordenadas X e Y consegue-se calcular a distância euclidiana de cada linha presente nas sessões. A ideia é, assim que essa distância alcançar um determinado ponto de corte (nomeamos de *"cut off"*) é realizada a quebra da sessão corrente em uma nova sessão. No exemplo da Tabela 5.11, realiza-se o corte na sessão "1506875407922788", que atingiu a distância de 41.5973501, acima do ponto de corte definido pelo experimento. No experimento, utilizou-se o valor 40 para o ponto de corte e, no exemplo, foi originada uma nova sessão.

Tabela 5.11: Corte em uma determinada sessão, Globo

	user_id	session_id	click_article_id	x_centroid	y_centroid	distance
0	15275	1506875407922788	101192	-23.93	-19.91	0.00
1	15275	1506875407922788	102738	4.27	-4.04	32.36
2	15275	1506875407922788	102701	4.27	-4.04	0.00
3	15275	1506875407922788	102692	16.82	-11.57	14.63
4	15275	1506875407922788_000	101193	-23.93	-19.91	**41.59**
5	15275	1506875407922788_000	102696	4.27	-4.04	32.36
6	15275	1506875407922788_000	102661	0.05	1.41	6.90
7	15275	1506875407922788_000	102686	4.27	-4.04	6.90
8	15275	1506875407922788_000	101194	-23.93	-19.91	32.36
9	15275	1506875407922788_000	102668	4.27	-4.04	32.36
10	15275	1506875407922788_000	102718	-18.14	3.25	23.58
11	15275	1506875407922788_000	102669	4.27	-4.04	23.58
12	15275	1506875407922788_000	101495	-23.93	-19.91	32.36
13	15275	1506875407922788_000	104173	4.27	-4.04	32.36
14	15275	1506889585128598	101195	-23.93	-19.91	32.36

Para os resultados obtidos com a base de dados do Lastfm, ilustrados na Figura 5.12, os cortes foram realizados quando atingiu-se um valor acima do ponto de corte estabelecido. Porém, quando comparados ao conjunto de dados da Globo, nota-se altos valores que representam a distância euclidiana.

Tabela 5.12: Corte em duas sessões, Lastfm

	user_id	track_name	x_centroid	y_centroid	session_id	distance
283	user_000001	The Riot Act	-46.88	-19.26	13_000	**62.06**
284	user_000001	Holding You	-10.51	47.93	13_000	76.40
285	user_000001	Standing Right Here	-4.16	11.51	13_000	36.96
286	user_000001	Free	17.37	27.52	13_000	26.83
287	user_000001	Stars In Your Eyes	-4.16	11.51	13_000	26.83
288	user_000001	Loran'S Dance	8.49	-23.41	14	37.15
289	user_000001	Early Morning	-46.88	-19.26	14_000	**55.53**
290	user_000001	Cash Flow	-4.16	11.51	14_000	52.65
291	user_000001	Microphone Master	37.33	13.64	14_000	41.55
292	user_000001	Cow	17.37	27.52	14_000	24.30
293	user_000001	Extra Ignored	34.88	-18.28	14_000	49.04
294	user_000001	Hibiya	29.46	-2.40	14_000	16.78
295	user_000001	Dead Death	17.37	27.52	14_000	32.27
296	user_000001	Lubumba '98	8.49	-23.41	14_000	51.70
297	user_000001	Face À La Mer	-4.16	11.51	14_000	37.15

Um outro exemplo, é quando não há nenhum corte atingido ou a sessão possui uma única interação. Nesse caso, a sessão permanece a mesma, sem alterações, conforme a Tabela 5.13.

Tabela 5.13: Sessão com id 17 e sem corte, Lastfm

	user_id	artist_name	track_name	x_centroid	y_centroid	session_id	distance
319	user_000001	Cornelius	Music	2.013273	42.734436	17	0.0

Nesta seção foi mostrada a etapa 1 do método FIP-SHA, corte em sessões. Foi escolhida a distância euclidiana, pois ela representa uma dissimilaridade entre os pontos, podendo realizar os cortes na sessão corrente. Na implementação atual, consegue-se realizar mais de um corte em uma conta, porém somente um corte por sessão (*session_id*). Um ponto a ser implementado no futuro, é a possibilidade de realizar múltiplos cortes em cada sessão. Um segundo item que nos chama atenção, é na construção do arquivo de *Word Embeddings*, podendo existir outras abordagens para a construção a serem exploradas no futuro. Um segundo experimento que realizamos 5.2.1 foi utilizando outro conjunto de dados do Lastfm com mais características presentes no conjunto, o que comprovou ser um influenciador para o resultado final.

5.2.1 Experimento 1 - conjunto de dados com *Tags*

O objetivo deste experimento é avaliar se as características presentes no conjunto de dados influenciam na construção do arquivo de *Word Embeddings* e logo, tem influência no resultado do algoritmo t-SNE, embora utilizando-se dados da mesma origem. Neste experimento, utiliza-se um segundo conjunto de dados do Lastfm que utiliza o conceito de *tags*.

Seis arquivos referentes a base de dados do Lastfm são fornecidos pelo GroupLens[5] em um dos workshops do HetRec, no ano de 2011, consistindo nos seguintes arquivos:

- artist.dat: contém informações sobre artistas, músicas ouvidas e marcadas pelos usuários;
- tags.dat: contém o conjunto de *tags* disponíveis no conjunto de dados;
- user_artists.dat: contém os artistas ouvidos por cada usuário. O arquivo também fornece uma contagem de escuta para cada tupla [usuário, artista];
- user_taggedartists.dat e user_taggedartists-timestamp.dat: contêm as atribuições de *tags* de artistas fornecidas por cada usuário. Os arquivos também contém a marcação de data e hora de quando as atribuições de *tags* foram concluídas;
- user_friends.dat: esse arquivo contém as relações de amizade entre os usuários no banco de dados.

Esse conjunto de dados é voltado para informações de *tags* e busca-se saber a similaridade do artista com essas informações. Portanto, utilizou-se os arquivos tags.dat e user_taggedartists.dat Tabelas 5.14 e 5.15.

Tabela 5.14: Arquivo tags.dat

	tagID	tagValue
0	1	metal
1	2	alternative metal
2	3	goth rock
3	4	black metal
4	5	death metal
...	...	...
11941	12644	suomi
11942	12645	symbiosis
11943	12646	sverige
11944	12647	eire
11945	12648	electro latino

Em análise preliminar, o arquivo user_friends.dat é o que parece ser menos útil.

[5]https://grouplens.org/datasets/hetrec-2011/

Tabela 5.15: Arquivo user_taggedartists.dat

	artistID	tagID
0	52	13
1	52	15
2	52	18
3	52	21
4	52	41
...	...	...
186474	16437	4
186475	16437	292
186476	16437	2087
186477	16437	2801
186478	16437	3335

Denotar alguém como amigo não se traduz em padrões de escuta semelhantes e conclui-se que é improvável que esses dados adicionem qualquer precisão aos modelos que foram construídos até então.

O conjunto representado na Figura 5.4 contém as informações das *tags* agrupadas por artista e associadas por usuário. A coluna *tagValue* foi utilizada como parâmetro para o *TfidfVectorizer* e gerou a matriz esparsa de dimensão 86608x7591 com 5.328.873 elementos.

Figura 5.4: Conjunto de dados da união de artista e *tags*, por usuário

	tagValue	id	name	userID	artistID	weight
0	j-rock,visual kei,gothic,japanese,weeabo,jrock...	1	MALICE MIZER	34	1	212
1	j-rock,visual kei,gothic,japanese,weeabo,jrock...	1	MALICE MIZER	274	1	483
2	j-rock,visual kei,gothic,japanese,weeabo,jrock...	1	MALICE MIZER	785	1	76
3	electronic,ambient,seen live,german,industrial...	2	Diary of Dreams	135	2	1021
4	electronic,ambient,seen live,german,industrial...	2	Diary of Dreams	257	2	152
...	...	...	...	...	...	...
86603	80s,alternative,electronica,noise,trip beat	18737	Ciccone Youth	454	18737	560
86604	electronic,trip-hop,rock,alternative,alternati...	18739	Apollo 440	454	18739	379
86605	ebm,industrial	18740	Die Krupps	454	18740	320
86606	experimental,dead music	18741	Diamanda Galás	454	18741	301
86607	chillout,downtempo,ambient,avant-garde,alterna...	18744	Oz Alchemist	454	18744	286

86608 rows × 6 columns

O arquivo de *Word Embeddings* gerado para o conjunto de dados que contém informação de *tags* está representado na Figura 5.5. O resultado obtido utilizando o arquivo como entrada para o algoritmo t-SNE, pode ser visualizado na Figura 5.6.

Quando analisa-se uma base de dados que possui a noção de *tags* o resultado é mais satisfatório do que a base originalmente utilizada e apresentada no capítulo 5.1. A justificativa encontra-se nas características presentes no conjunto e aparentam ser mais relevantes, ajudaram-nos na construção das etapas até gerar o arquivo de entrada para o t-SNE.

Figura 5.5: Arquivo de Embeddings, GroupLens

```
array([[0.00068591, 0.00112813, 0.00136666, 0.        , 0.        ,
        0.        ],
       [0.00068591, 0.00112813, 0.00136666, 0.        , 0.        ,
        0.        ],
       [0.00068591, 0.00112813, 0.00136666, 0.        , 0.        ,
        0.        ],
       ...,
       [0.        , 0.00141267, 0.00041736, 0.00043899, 0.        ,
        0.00033659],
       [0.        , 0.00053505, 0.00308773, 0.00171337, 0.        ,
        0.00110954],
       [0.        , 0.00103561, 0.00082523, 0.0044352 , 0.        ,
        0.0096871 ]])
```

Figura 5.6: Resultado do tsne, GroupLens

```
array([[20.076683  , -8.084514  ],
       [20.076683  , -8.084514  ],
       [20.076683  , -8.084514  ],
       ...,
       [21.22815   , -9.026172  ],
       [11.187972  , -0.21945494],
       [29.806145  , 14.810456  ]], dtype=float32)
```

5.3 Experimento 2 - parametrização do algoritmo t-SNE

O objetivo deste experimento é avaliar o algoritmo t-SNE atribuindo diferentes valores para o parâmetro *perplexity*. Para realizar esse experimento foi utilizado como base o estudo realizado por Wattenberg (MAATEN; HINTON, 2008b) que ilustra o desempenho do t-SNE parametrizado com diferentes valores e diante de alguns conjuntos de dados, mostrando alterações finais quando altera-se o mesmo. Um dos valores mais importantes do algoritmo t-SNE é o parâmetro de *perplexity*, o qual equilibra a atenção que o t-SNE dá aos aspectos locais e globais dos dados e pode ter efeito no gráfico resultante. Por isso, diferentes valores foram testados, nas bases de dados da Globo, Figura 5.7 e do Lastfm, Figura 5.8. O valor padrão para o t-SNE, se não configurado nenhum parâmetro é *perplexity* igual a 30. Como resultado do experimento, a ordem dos conjuntos de imagens a seguir apresenta os valores do parâmetro *perplexity* 5, 2, 30 e 50, respectivamente.

Pode-se observar que conforme o valor de *perplexity* aumenta, obtemos uma configuração cada vez mais estável. t-SNE faz parte de um processo iterativo em que as diferenças entre as amostras são continuamente refinadas. Pode-se definir também um limite para o número máximo de iterações a serem realizadas. Essa abordagem é bastante utilizada para conjuntos de dados grandes, acelerando o tempo necessário para obter uma

Figura 5.7: Resultado t-SNE com valor de *perplexity* variado - Globo

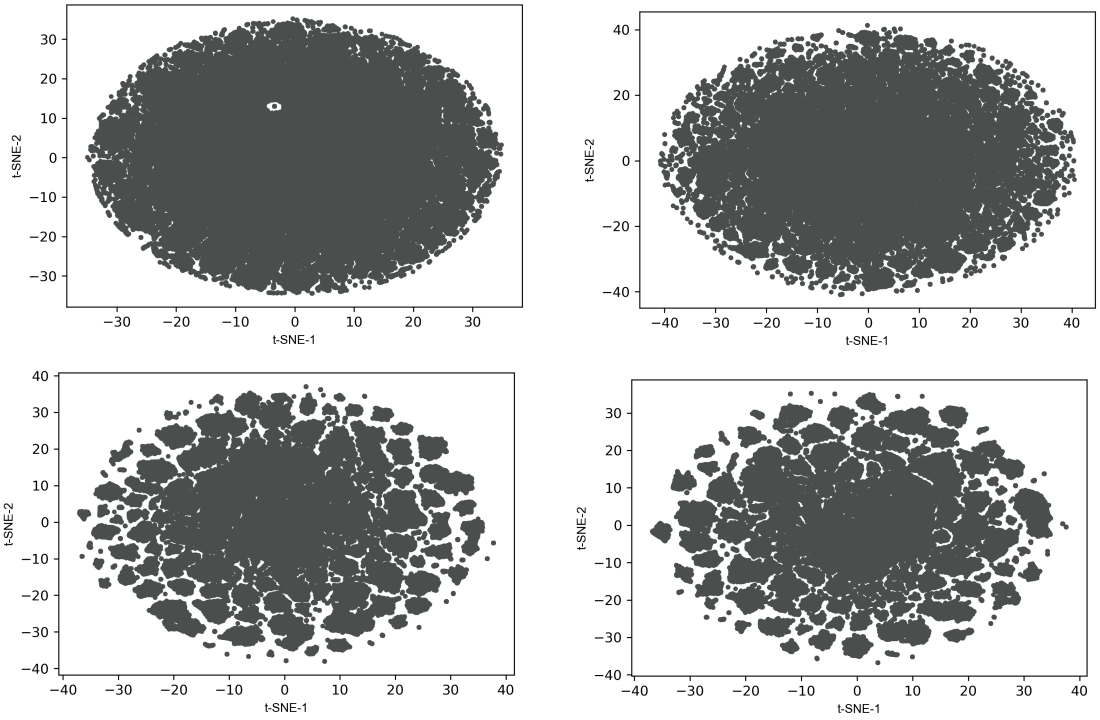
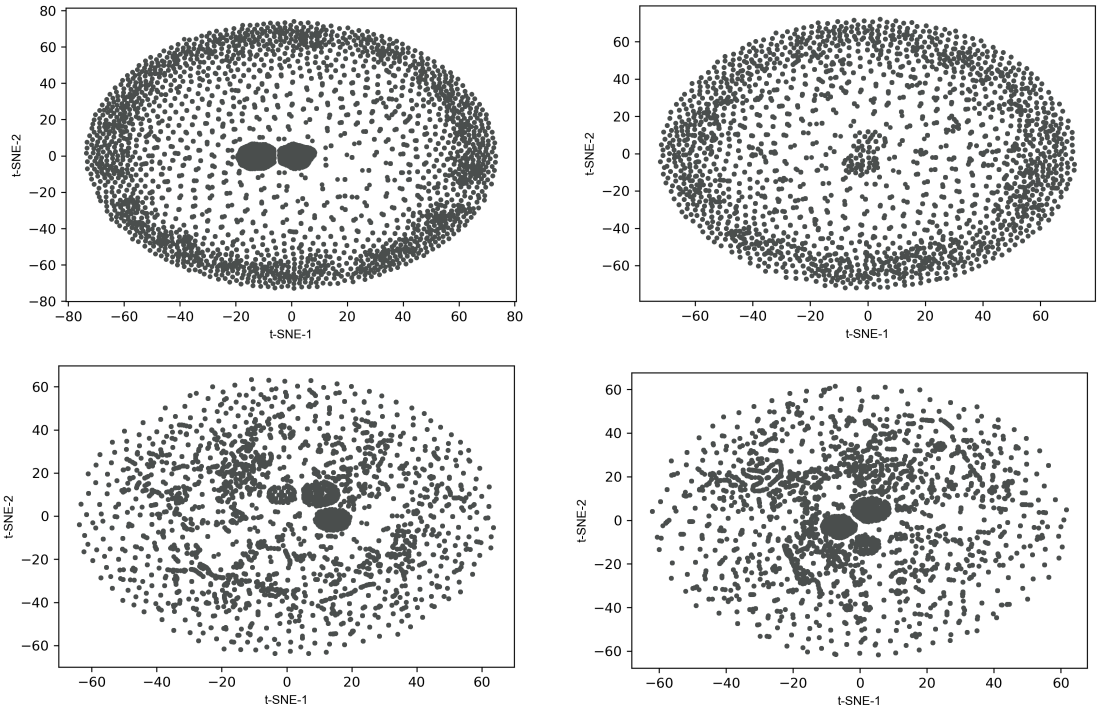


Figura 5.8: Resultado t-SNE com valor de *perplexity* variado - Lastfm



resposta.

Não existe uma regra específica para determinar os valores absolutos dos hiperparâmetros como *perplexity* e número de iterações, depende do conjunto de dados. Portanto, para saber quais valores utilizar, tem-se que executar o algoritmo t-SNE repetidamente

com diferentes valores antes de chegar ao resultado que melhor se adéque.

5.4 Experimento 3 - calibragem do parâmetro *Damping*, e desempenho do algoritmo Affinity Propagation

O objetivo deste experimento é avaliar o desempenho do algoritmo *Affinity Propagation* na escolha do valor para o parâmetro *damping* e, ao final do experimento, escolher o valor apropriado para cada um dos conjuntos de dados. O algoritmo *Affinity Propagation* precisa obter o centro de agrupamento por meio de iteração contínua, o que leva à alta complexidade de tempo do algoritmo. Os agrupamentos e os exemplos retornados são obtidos com a influência do parâmetro *damping*. Esse valor é responsável por manter um equilíbrio entre convergência e oscilação, ou seja, pode-se dizer que esse parâmetro controla a convergência e a velocidade do algoritmo.

O valor do *damping* deve ser maior ou igual a 0.5 e menor do que 1, sendo o valor 0.5 o valor padrão. Um valor maior não significa ficar preso em oscilações, mas reduzir a taxa de convergência, enquanto um valor menor resulta em uma taxa rápida de convergência, mas com um risco de não convergência quando o procedimento de passagem de mensagens é encerrado. Para se saber qual é o valor mais adequado para as bases de dados da Globo e do Lastfm, foram realizados experimentos com uma amostragem das 5 primeiras contas e com diferentes valores de *damping* e os resultados comparados.

O fator de amortecimento afeta o número de agrupamentos, como pode ser visto nas Figuras 5.9 e 5.10. O fator de amortecimento também desempenha um papel decisivo na valocidade de convergência do algoritmo. A escolha inadequada desse valor pode levar à oscilação do algoritmo, tornando fácil a sua não convergência e, por fim, influenciando no efeito final do agrupamento. Como resultado dos experimentos realizados, o valor de *damping* 0.9 apresentou um desempenho melhor com relação aos outros valores testados em ambas as bases de dados, o que nos fez optar por esse valor no decorrer dos próximos experimentos realizados.

As subseções 5.4.1 e 5.4.2 apresentam uma análise de desemepenho realizada com base no número de iterações que ocorreram de modo a produzir o número ideal de grupos.

Figura 5.9: Globo, quantidade de agrupamento gerados por valor de *damping*

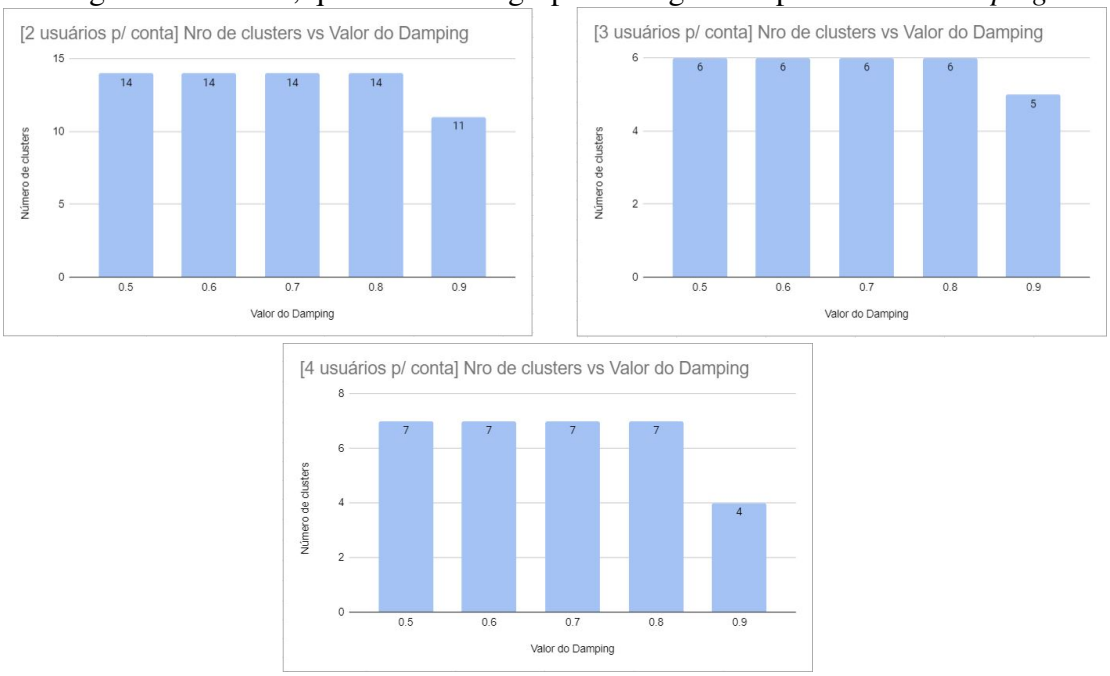
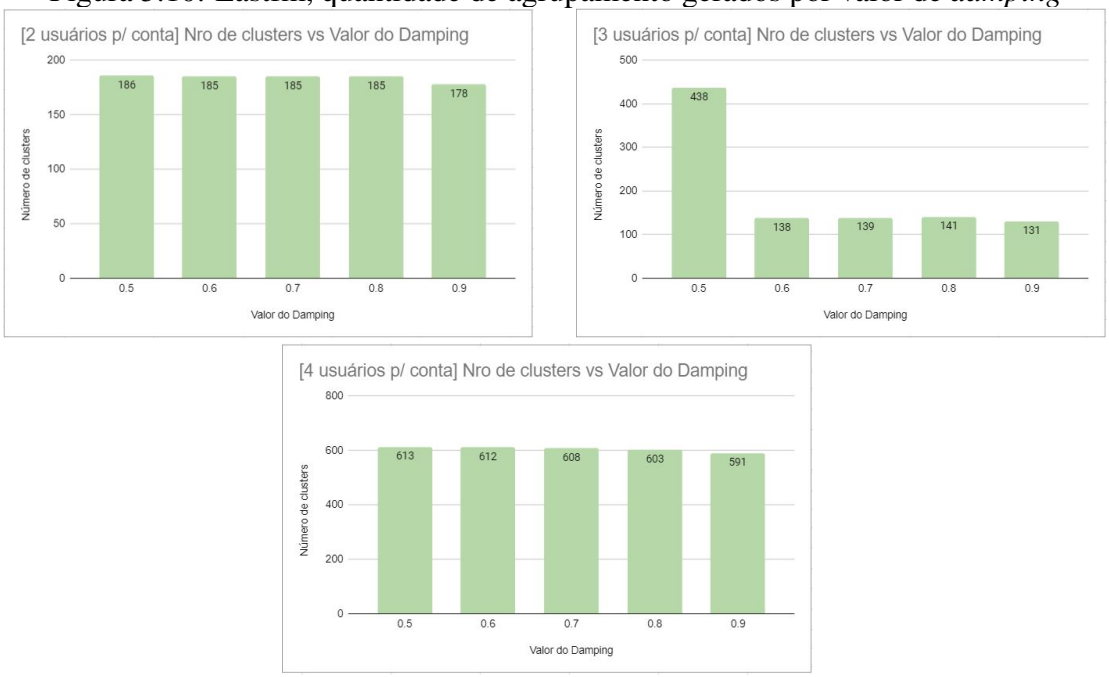


Figura 5.10: Lastfm, quantidade de agrupamento gerados por valor de *damping*



5.4.1 Experimento 3 - Resultados Globo

Nesse experimento, a quantidade de agrupamentos e o número de iterações foram comparados e o valor do parâmetro *damping* analisado. Os resultados deste estudo são apresentados nas Tabelas 5.16, 5.17 e 5.18 para a base de dados da Globo.

Analisando os resultados, após utilizar dados simulados para uma conta que con-

tém 2 usuários, o valor mínimo de *damping* para a conta com índice 0 resultou em 14 agrupamentos, enquanto o *damping* com o valor mais alto para a mesma conta resultou em 11 agrupamentos. Outro exemplo, é a conta de índice 2 e valor mínimo do *damping* resultou em 4 agrupamentos e um valor maior resultou em 2 agrupamentos.

Tabela 5.16: Performance do Algoritmo *Affinity Propagation* baseado no parâmetro *damping* - conta com 2 usuários, Globo

conta	numero de usuários	agrupamentos	iterações	damping
0	2	14	78	0.5
0	2	14	72	0.6
0	2	14	97	0.7
0	2	14	142	0.8
0	2	**11**	16	0.9
1	2	5	20	0.5
1	2	5	22	0.6
1	2	5	25	0.7
1	2	5	32	0.8
1	2	5	52	0.9
2	2	4	41	0.5
2	2	4	50	0.6
2	2	4	66	0.7
2	2	4	45	0.8
2	2	**2**	38	0.9
3	2	4	61	0.5
3	2	4	78	0.6
3	2	4	106	0.7
3	2	4	161	0.8
3	2	5	44	0.9

O mesmo é possível visualizar nas tabelas 5.17 e 5.18, no qual compara-se a quantidade de agrupamentos e iterações com o parâmetro *damping*. Para ambos os cenários de contas com 3 e 4 usuários, o valor que teve o melhor desempenho foi 0.9.

Tabela 5.17: Performance do Algoritmo *Affinity Propagation* baseado no parâmetro *damping*- conta com 3 usuários, Globo

conta	numero de usuários	agrupamentos	iterações	damping
0	3	6	200	0.5
0	3	6	27	0.6
0	3	6	32	0.7
0	3	6	42	0.8
0	3	5	56	0.9
1	3	5	38	0.5
1	3	5	45	0.6
1	3	5	58	0.7
1	3	4	49	0.8
1	3	4	37	0.9
2	3	7	49	0.5
2	3	7	62	0.6
2	3	7	81	0.7
2	3	7	33	0.8
2	3	6	37	0.9
3	3	7	26	0.5
3	3	7	28	0.6
3	3	7	32	0.7
3	3	7	41	0.8
3	3	7	69	0.9

Tabela 5.18: Performance do Algoritmo *Affinity Propagation* baseado no parâmetro *damping* - conta com 4 usuários, Globo

conta	numero de usuários	agrupamentos	iterações	damping
0	4	8	32	0.5
0	4	8	27	0.6
0	4	8	28	0.7
0	4	8	36	0.8
0	4	8	56	0.9
1	4	7	64	0.5
1	4	7	82	0.6
1	4	7	112	0.7
1	4	7	171	0.8
1	4	**4**	43	0.9
2	4	10	21	0.5
2	4	10	23	0.6
2	4	10	28	0.7
2	4	10	36	0.8
2	4	10	61	0.9
3	4	7	61	0.5
3	4	7	78	0.6
3	4	7	105	0.7
3	4	7	160	0.8
3	4	9	46	0.9

5.4.2 Experimento 3 - Resultados Lastfm

Nesta subseção, apresenta-se os resultados da base de dados do Lastfm. Foram realizados testes com valor de *damping* variado, observando a quantidade de agrupamentos e o número de iterações executadas. Os resultados são apresentados nas Tabelas 5.19, 5.20 e 5.21. Vale lembrar que para a base de dados do Lastfm é esperada uma grande quantidade de agrupamentos, devido ao excessivo volume de dados por conta.

Tabela 5.19: Performance do Algoritmo *Affinity Propagation* baseado no parâmetro *damping* - conta com 2 usuários, Lastfm

conta	numero de usuários	agrupamentos	iterações	damping
0	2	186	86	0.5
0	2	185	93	0.6
0	2	185	122	0.7
0	2	185	184	0.8
0	2	**178**	200	0.9
1	2	439	115	0.5
1	2	436	112	0.6
1	2	436	170	0.7
1	2	434	200	0.8
1	2	**421**	200	0.9
2	2	406	85	0.5
2	2	399	83	0.6
2	2	399	113	0.7
2	2	399	171	0.8
2	2	**391**	200	0.9
3	2	234	62	0.5
3	2	238	106	0.6
3	2	238	107	0.7
3	2	238	162	0.8
3	2	**229**	200	0.9

Similar ao experimento que foi realizado para a base de dados da Globo, o valor do parâmetro *damping* que performou melhor para as contas que possuem 2, 3 e 4 usuários, foi o valor de 0.9 e assim, será utilizado no decorrer dos demais experimentos.

Tabela 5.20: Performance do Algoritmo *Affinity Propagation* baseado no parâmetro *damping* - conta com 3 usuários, Lastfm

conta	numero de usuários	agrupamentos	iterações	damping
0	3	438	200	0.5
0	3	138	85	0.6
0	3	139	123	0.7
0	3	141	187	0.8
0	3	131	145	0.9
1	3	596	84	0.5
1	3	592	84	0.6
1	3	592	120	0.7
1	3	592	173	0.8
1	3	587	200	0.9
2	3	291	200	0.5
2	3	291	115	0.6
2	3	290	125	0.7
2	3	290	188	0.8
2	3	274	200	0.9
3	3	694	200	0.5
3	3	697	200	0.6
3	3	703	200	0.7
3	3	694	186	0.8
3	3	645	200	0.9

Tabela 5.21: Performance do Algoritmo *Affinity Propagation* baseado no parâmetro *damping* - conta com 4 usuários, Lastfm

conta	numero de usuários	agrupamentos	iterações	damping
0	4	613	98	0.5
0	4	612	117	0.6
0	4	608	150	0.7
0	4	603	200	0.8
0	4	591	200	0.9
1	4	640	200	0.5
1	4	634	156	0.6
1	4	629	117	0.7
1	4	630	173	0.8
1	4	618	200	0.9
2	4	740	200	0.5
2	4	739	200	0.6
2	4	748	146	0.7
2	4	751	200	0.8
2	4	687	200	0.9

5.5 Experimento 4 - avaliação dos agrupamentos

Este experimento avalia a performance do algoritmo que agrupa as sessões no cenário de contas compartilhadas que possuem 2, 3 e 4 usuários. Os principais objetivos são: i) avaliar a performance do algoritmo de agrupaento; e ii) analisar os resultados utilizando as métricas definidas. Esse resultado inicial é importante porque mostra que o uso do algoritmo de agrupamento *Affinity Propagation* é capaz de selecionar grupos em quantidades razoáveis sem a necessidade de que usuários tenham que intervir manualmente para conduzir o processo.

5.5.1 Metodologia

Em cada conjunto de dados foram geradas as contas compartilhadas com 2, 3 e 4 usuários por conta e, para cada conta gerada, foi aplicada a métrica de similaridade por cosseno entre as sessões. Por fim, a análise dos resultados baseou-se nas métricas de avaliação definidas em *avaliação de desempenho de agrupamento*[6], as quais são as mesmas métricas utilizadas no trabalho *baseline* SHE-UI, seção 3.3. As próximas subseções especificam, em detalhes, as métricas utilizadas para avaliar os resultados este experimento.

5.5.1.1 Adjusted Rand index (ARI)

A métrica ARI calcula uma medida de similaridade entre dois agrupamentos considerando todos os pares de amostras e contando pares que são atribuídos no mesmo agrupamento ou em agrupamentos diferentes (chamados de agrupamentos preditos ou verdadeiros). A rotulagem perfeita é avaliada em 1 e rotulações ruins têm valores negativos ou próximos de 0.

$$ARI = \frac{RI - Expected\_RI}{max(RI) - Expected\_RI} \tag{5.1}$$

O resultado do agrupamento geralmente é comparado com a verdade "fundamental" para avaliar a precisão. Ou seja, métrica ARI é uma medida de concordância de similaridade popular entre duas partições. A preferência estimada na métrica ARI deve ser tratada como a preferência ideal porque os *scores* da métrica ARI são avaliados usando *labels* ditas como "verdadeiras".

5.5.1.2 Normalized Mutual Information (NMI)

A métrica NMI é a normalização da métrica *Mutual Information* (MI). MI mede a concordância das duas atribuições, ignorando as permutações, é uma medida simétrica menor ou igual a 1. Valores próximos a 0 indicam que os rótulos são independentes, enquanto valores próximos a 1 indicam uma concordância significativa entre dois agrupamentos. A rotulagem aleatória tem um MI negativo e para utilizar a métrica requer-se o conhecimento da verdadeira rotulagem.

[6]https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

NMI é definido com base na fórmula a seguir:

$$NMI(U,V) = \frac{MI(U,V)}{mean(H(U),H(V))} \tag{5.2}$$

Valores próximos a 0 indicam que os rótulos são independentes, enquanto valores próximos a 1 indicam uma concordância significativa entre dois agrupamentos. Além disso, a rotulagem perfeita tem um NMI de 1, e a rotulagem aleatória tem um NMI negativo.

5.5.1.3 Adjusted Mutual Information (AMI)

A métrica AMI é a segunda variação da métrica MI. AMI é uma técnica mais recente do que a NMI e é, por usa vez, ajustada e definida pela fórmula:

$$AMI(U,V) = \frac{MI(U,V) - E\{MI(U,V)\}}{max\{H(U),H(V)\} - E\{MI(U,V)\}} \tag{5.3}$$

Esta é novamente uma medição simétrica e seu limite está em 1. A rotulagem perfeita tem um AMI de 1 e a rotulagem aleatória tem um AMI próximo de 0. Valores próximos a 0 indicam que os rótulos são amplamente independentes, quando valores próximos a 1 indicam uma concordância significativa entre os dois agrupamentos.

5.5.1.4 Fowlkes-Mallows (FMI)

A métrica FMI indica o grau de similaridade entre dois agrupamentos. FMI é a métrica dita como a média geométrica da precisão e revocação. O FMI pode ser calculado quando se sabe a classificação dos itens que estão no agrupamento.

$$FMI = \frac{TP}{\sqrt{(TP+FP)(TP+FN)}} \tag{5.4}$$

Onde:

- TP é o número de verdadeiros positivos, ou seja, o número de pares de pontos no mesmo agrupamento na rotulagem verdadeira e prevista;
- FP é o número de falsos positivos, ou seja, pares com a mesma rotulação verdadeira, mas em agrupamentos previstos diferentes;
- FN é o número de falsos negativos, ou seja, pares nos mesmos agrupamentos previstos, mas com rotulações verdadeiras diferentes.

Um agrupamento previsto terá um FMI de 1, enquanto um agrupamento independente das classes reais terá próximo de 0.

5.5.2 Resultados

A seguir estão detalhados os resultados obtidos pelo método que agrupa as sessões em perfis de usuários presentes nas contas compartilhadas para as bases de dados Globo e Lastfm.

5.5.2.1 Resultados base de dados da Globo

Primeiramente, analisam-se dois usuários aleatórios, por exemplo, usuário *15275* com 746 relacionamentos entre sessões e categorias e o usuário *123929* com 29 relacionamentos. As Tabelas 5.22 e 5.23 apresentam as categorias e sessões pertencentes ao usuário 15275 e ao usuário 123929, respectivamente. O usuário *15275* contém registros somente das categorias 226 e 228, já o usuário *123929* contém registros de outras categorias em suas sessões, representando a interação com mais categorias de notícias.

Tabela 5.22: Categorias nas sessões pertencentes ao usuário 15275

	user_id	session_id	category_id	account_id
0	15275	1506875407922788	226	0.0
1	15275	1506875407922788	226	0.0
2	15275	1506875407922788	226	0.0
3	15275	1506875407922788	226	0.0
4	15275	1506875407922788	226	0.0
..	..	..	..	..
741	15275	1508123293257891	228	0.0
742	15275	1508123293257891	228	0.0
743	15275	1508123293257891	228	0.0
744	15275	1508123293257891	228	0.0
745	15275	1508123293257891	228	0.0

As Tabelas a seguir apresentam os resultados do agrupamento após executar o algoritmo *Affinity Propagation* com valor de *damping* 0,9. Pode-se observar na Tabela 5.24, que raramente as categorias se repetem entre os grupos, sendo um bom indício de resultado do agrupamento. Somente o exemplo do agrupamento de número 7 se deu erroneamente, e as sessões pertencentes a esse grupo deveriam fazer parte do agrupamento de número 4, por conterem as mesmas categorias. Quando compara-se com o agrupamento quando o valor de *damping* é o valor padrão em 0,5, existem agrupamentos com categorias repetidas, é o caso dos agrupamentos *11, 4, 1 e 10*, ilustrados na Tabela 5.25.

A Tabela 5.26 apresenta os resultados das contas analisadas, quando utilizou-se um conjunto de dados com somente 2 usuários como dados de entrada. O valor estimado

Tabela 5.23: Categorias nas sessões pertencentes ao usuário 123929

	user_id	session_id	category_id	account_id
746	123929	1507031176212612	412	0.0
747	123929	1507031176212612	418	0.0
748	123929	1507031176212612	209	0.0
749	123929	1507031176212612	399	0.0
750	123929	1507562453117515	418	0.0
751	123929	1507562453117515	317	0.0
752	123929	1507562453117515	254	0.0
753	123929	1507562453117515	289	0.0
754	123929	1507569787756787	418	0.0
755	123929	1507569787756787	437	0.0
756	123929	1507569787756787	442	0.0
757	123929	1507569787756787	429	0.0
758	123929	1507399040959184	399	0.0
759	123929	1507399040959184	247	0.0
760	123929	1507230463232653	375	0.0
761	123929	1507230463232653	250	0.0
762	123929	1507230463232653	434	0.0
763	123929	1507391808207631	281	0.0
764	123929	1507391808207631	281	0.0
765	123929	1507391808207631	437	0.0
766	123929	1507926740328318	281	0.0
767	123929	1507926740328318	437	0.0
768	123929	1507926740328318	455	0.0
769	123929	1507926740328318	353	0.0
770	123929	1507548333183379	437	0.0
771	123929	1507548333183379	437	0.0
772	123929	1507906094135594	437	0.0
773	123929	1507906094135594	147	0.0
774	123929	1507906094135594	435	0.0

Tabela 5.24: agrupamentos com valor de damping 0.9

grupo/perfil	categorias
0	412, 418, 209, 399
6	418, 317, 254, 289
8	418, 437, 442, 429
10	281, 437, 455, 353
1	375, 250, 434
2	281, 281, 437
9	437, 147, 435
4	226, 228
3	399, 247
7	226, 228
5	437

Tabela 5.25: agrupamentos com valor de damping 0.5

grupo/perfil	categorias
0	412, 418, 209, 399
7	418, 317, 254, 289
9	418, 437, 442, 429
13	281, 437, 455, 353
2	375, 250, 434
3	281, 281, 437
12	437, 147, 435
11	226, 228
4	226, 228
1	226, 228
10	226, 228
5	399, 247
6	437
13	226

do ARI é acima de 0,5. As demais métricas resultaram em valores acima do esperado, a média do NMI ficou em 0,8, o AMI mais baixo foi 0,598 e o FMI 0,812.

Tabela 5.26: Avaliação das métricas de agrupamento para as contas com 2 usuários

	Agrupamentos	ARI	NMI	AMI	FMI
0	11	0,926	0,939	0,876	0,969
1	5	0,755	0,845	0,771	0,812
2	2	0,800	0,752	0,726	0,901
3	5	0,581	0,755	0,598	0,691

Tabela 5.27: Avaliação das métricas de agrupamento para as contas com 3 usuários

	Agrupamentos	ARI	NMI	AMI	FMI
0	5	0,922	0,867	0,829	0,954
1	4	0,926	0,928	0,915	0,945
2	6	0,882	0,909	0,864	0,902
3	7	0,778	0,836	0,773	0,816

5.5.2.2 Resultados base de dados do Lastfm

Para a base de dados do Lastfm, a análise é similar, porém, devido à natureza e o volume dos dados por conta compartilhada, a quantidade de grupos/perfis foi bem maior, o que era esperado nesses casos. Analisou-se uma escolha de conta que possui 2 usuários, representada pela Figura 5.11, possuindo valores únicos de *track_name* (dado utilizado para o agrupamento) com aproximadamente 9 mil dados únicos.

Figura 5.11: exemplo de conta com 2 usuários

	user_id	session_id	track_name	account_id
0	user_000666	597228	Czarnuch	0.0
1	user_000666	597228	Iv L.O. '98	0.0
2	user_000666	597228	King	0.0
3	user_000666	597496	Czarnuch	0.0
4	user_000666	597496	Kombajn Bizon	0.0
...	...	...	...	...
19131	user_000735	658620	Stolte Nordmenn	0.0
19132	user_000735	658620	Deg Og Meg Åneil	0.0
19133	user_000735	658619	King Of The Mountain Road Master	0.0
19134	user_000735	658656	P320080605-Emsmith	0.0
19135	user_000735	658661	Ode To...	0.0

O usuário *user_000666* possui 537 sessões únicas e o usuário *user_000735* possui 333 sessões únicas com diferentes músicas escutadas por sessão. As sessões e suas relações por usuário são representadas nas Tabelas 5.29 e 5.30.

Tabela 5.28: Avaliação das métricas de agrupamento para as contas com 4 usuários

	Agrupamentos	**ARI**	**NMI**	**AMI**	**FMI**
0	8	0,881	0,904	0,869	0,914
1	4	0,771	0,735	0,682	0,858
2	10	0,968	0,980	0,966	0,972
3	9	0,789	0,893	0,838	0,816

Tabela 5.29: Categorias nas sessões pertencentes ao usuário user_000666

	user_id	session_id	track_name	account_id
0	user_000666	597228	Czarnuch	0.0
1	user_000666	597228	Iv L.O. '98	0.0
2	user_000666	597228	King	0.0
3	user_000666	597496	Czarnuch	0.0
4	user_000666	597496	Kombajn Bizon	0.0
..	..	..	..	..
11398	user_000666	597727	Jus' Reach	0.0
11399	user_000666	597737	Out Here We Are Stoned (X-Dream Remix)	0.0
11400	user_000666	597737	Friagram	0.0
11401	user_000666	597741	Cut It Loose	0.0
11402	user_000666	597764	Save Your Love	0.0

Tabela 5.30: Categorias nas sessões pertencentes ao usuário user_000735

	user_id	session_id	track_name	account_id
11403	user_000735	658346	Ride	0.0
11404	user_000735	658346	I Love You	0.0
11405	user_000735	658346	Unglued	0.0
11406	user_000735	658346	Army Ants	0.0
11407	user_000735	658346	Crackerman	0.0
..	..	..	..	..
19131	user_000735	658620	Stolte Nordmenn	0.0
19132	user_000735	658620	Deg Og Meg Ånei!	0.0
19133	user_000735	658619	King Of The Mountain Road Master	0.0
19134	user_000735	658656	P320080605-Emsmith	0.0
19135	user_000735	658661	Ode To...	0.0

A Tabela 5.31 apresenta os resultados dos casos analisados, para o cenário de 2 usuários por conta. Os resultados de ARI foram acima de 0,7 e métricas como AMI acima de 0,8.

Tabela 5.31: Avaliação das métricas de agrupamento para as contas com 2 usuários

	Agrupamentos	**ARI**	**NMI**	**AMI**	**FMI**
0	178	0,946	0,989	0,962	0,946
1	421	0,783	0,960	0,870	0,791
2	391	0,884	0,980	0,929	0,886
3	229	0,796	0,966	0,882	0,805

Analisando uma conta que possui 3 usuários, Figura 5.12, com 78.700 entradas de log. O usuário user_000857 possui 4.934 *track_name* únicos, o usuário user_000120 possui 6.435 e o usuário user_000963 possui 12.165. A Tabela 5.32 apresenta os resultados das métricas de agrupamento para esse exemplo.

Uma conta contendo 4 usuários é demonstrada na Figura 5.13, conta essa que possui 45.787 entradas de log, sendo 4.073 para o usuário user_000954, 3.755 para o usuário user_000839, 2.637 para o usuário user_000900 e 3.765 para o usuário user_000491, valores esses de *track_name* únicos relacionados aos usuários da conta. A Tabela 5.33

Figura 5.12: exemplo de conta com 3 usuários

	user_id	session_id	track_name	account_id
0	user_000857	782847	Everywhere With Helicopter	3.0
1	user_000857	782847	Full On Idle	3.0
2	user_000857	782847	Let Me Stand Next To Your Flower (Live)	3.0
3	user_000857	782847	Brain Damage	3.0
4	user_000857	782847	Brain Damage	3.0
...	...	...	...	...
78695	user_000963	880189	Fead An Iolar	3.0
78696	user_000963	880189	The Mole Man Of Hackney	3.0
78697	user_000963	880194	Cut Out Girl Scout	3.0
78698	user_000963	880196	The Faucets Are Dripping	3.0
78699	user_000963	880207	Kenny'S Sound	3.0

Tabela 5.32: Avaliação das métricas de agrupamento para as contas com 3 usuários

	Agrupamentos	**ARI**	**NMI**	**AMI**	**FMI**
0	421	0,742	0,953	0,842	0,753
1	587	0,899	0,985	0,941	0,901
2	274	0,760	0,961	0,862	0,772
3	645	0,971	0,994	0,978	0,971

apresenta os resultados das métricas que avaliam os agrupamentos para esse exemplo.

Figura 5.13: exemplo de conta com 4 usuários

	user_id	session_id	track_name	account_id
0	user_000954	866475	Son Et Lumiere	4.0
1	user_000954	866475	Inertiatic Esp	4.0
2	user_000954	866475	Velvety Instrumental Version	4.0
3	user_000954	866475	Moon In The Bathroom	4.0
4	user_000954	866475	Alms	4.0
...	...	...	...	...
45782	user_000491	451183	The Nature (Feat. Justin Timberlake)	4.0
45783	user_000491	451183	Beeper	4.0
45784	user_000491	451183	One, Two One	4.0
45785	user_000491	451183	Big Things Poppin'	4.0
45786	user_000491	451183	Bring Em Out (Ext Intro)	4.0

Tabela 5.33: Avaliação das métricas de agrupamento para as contas com 4 usuários

	Agrupamentos	**ARI**	**NMI**	**AMI**	**FMI**
0	591	0,804	0,968	0,884	0,811
1	618	0,840	0,976	0,906	0,845
2	687	0,923	0,991	0,960	0,924

5.5.3 Análise da separação dos usuários

A Tabela 5.34 apresenta os resultados da média ponderada para o conjunto de dados da Globo. A maioria dos agrupamentos gerados alcançou uma separação de usuário acima de 0,60, valor 0,84 considerando a média simples e 0,91 considerando a média ponderada. Houveram alguns casos em que a separação do usuário apresentou resultado abaixo de 0,4, principalmente pela dificuldade da etapa de agrupamento em separar os usuários com o mínimo de metadados de usuário ou item disponíveis.

Tabela 5.34: Análise de separação de usuários para contas compartilhadas da Globo

Conta #	N. usuários	Média	D. Padrão	Média Ponderada
1	2	1	0	1
2	2	0,67	0,04	0,67
3	2	0,46	0,02	0,59
4	2	0,84	0,21	0,66
1	3	0,84	0,21	0,91
2	3	0,67	0,11	0,63
3	3	0,65	0,16	0,64
4	3	0,59	0,16	0,65
1	4	0,71	0,26	0,82
2	4	0,63	0,32	0,48
3	4	0,64	0,15	0,63
4	4	0,62	0,19	0,64

A Tabela 5.35 apresenta os resultados de separação dos usuários obtidos para o conjunto de dados do Lastfm. Pode-se observar que os resultados chegaram a 0,9, indicando uma eficácia alta em revelar os usuários por trás das contas compartilhadas construídas para a avaliação.

Tabela 5.35: Análise de separação de usuários para contas compartilhadas da Lastfm

Conta #	N. usuários	Média	D. Padrão	Média Ponderada
1	2	0,91	0,13	0,9
2	2	0,96	0,08	0,96
3	2	0,97	0,08	0,98
1	3	0,93	0,13	0,94
2	3	0,94	0,11	0,95
3	3	0,93	0,12	0,94
1	4	0,9	0,15	0,91
2	4	0,93	0,13	0,93
3	4	0,91	0,14	0,91

Em resumo, esses resultados fornecem evidências da eficácia da similaridade dos itens do usuário para a quebra das sessões online e o uso do método de agrupamento para agrupar essas sessões em perfis de usuário.

6 CONCLUSÃO

Pesquisas encontradas na literatura revelam-se capazes de revelar os perfis de usuários por trás de contas compartilhadas. No entanto, as soluções existentes dependem de uma grande quantidade de metadados (do usuário ou item) e podem depender de ambientes que lidem apenas com itens homogêneos (ou seja, itens de um único tipo) para ter um desempenho satisfatório. No método FIP-SHA, adota-se a abordagem para revelar perfis de usuários em contas compartilhadas considerando itens heterogêneos onde poucos metadados de itens são disponibilizados. FIP-SHA possui três etapas. A primeira consiste em identificar as sessões *online* dos usuários através. Tal é feito a partir da captura das ações de um usuário em um determinado instante e identificação de quando uma sessão começa e termina, representando assim sessões *online* dos usuários. A segunda etapa consiste em representar as sessões através dos termos e suas frequências que descrevem os itens visitados. Por fim, a terceira etapa consiste em agrupar as sessões em perfis de usuários.

O FIP-SHA foi avaliado utilizando-se de um conjunto de experimentos que avaliam as etapas do método e um último experimento responsável por avaliar o resultado do agrupamento das sessões que representam os perfis dos usuários em uma conta compartilhada. Para realizar os experimentos, foram construídos dados sintéticos que representam contas compartilhadas, através de duas bases de dados reais (Globo.com e Last.fm). Os experimentos fornecem evidências da viabilidade das etapas apresentadas no método FIP-SHA. Os experimentos relacionados às primeira e terceira etapas do método FIP-SHA (corte e agrupamento de sessões, respectivamente) apresentaram resultados satisfatórios. Durante os experimentos da primeira etapa observa-se que os dados presentes em cada conjunto de dados possuem grande influência no resultado final. Essa evidência comprova-se ao validar a solução utilizando um conjunto de dados com mais informações e o resultado final apresentou uma melhoria. Ao comparar os resultados com os trabalhos encontrados na literatura, FIP-SHA mostrou-se eficaz na identificação das sessões dos usuários através da similaridade dos itens para a quebra das sessões e no uso do método de agrupamento das sessões que resultam nos perfis de usuários.

Pode-se destacar limitações para o FIP-SHA, como realizar mais de um corte em uma sessão de uma conta compartilhada. Um segundo estudo para o futuro, são alternativas para a técnica de redução de dimensionalidade, além da necessidade de realizar-se um teste A/B para garantir a eficácia da implementação em conjunto com um sistema de

recomendação.

Como principal contribuição do presente trabalho, observa-se que o FIP-SHA apresenta soluções para as questões que estão aberto encontradas no estado da arte, mostrando ser capaz de realizar a identificação de usuários em contas compartilhadas ao considerar a quebra de sessão por análise da similaridade ao invés dos 30 minutos de inatividade. Comprova-se também o uso do algoritmo *Affinity Propagation* na originação dos agrupamentos que representam os perfis dos usuários. Mais importante, o FIP-SHA mostrou-se capaz de lidar com contas compartilhadas em um contexto em que os dados sobre as ações dos usuários (e itens visitados) são completamente anonimizados. Trata-se de um avanço importante em relação ao estado da arte, que depende de dados de entrada não anonimizados, acompanhados de metadados.

Trabalhos de conclusão emergiram da presente pesquisa, apresentando resultados promissores e contribuições acadêmicas para a literatura. Pedro Nerung responsabilizou-se pelo desenvolvimento do *Baseline*, a partir dos resultados viabilizou-se o compartativo do método FIP-SHA com o método SHE-UI. Matheus Tura contribui com a análise e auxilio no desenvolvimento da primeira etapa do método FIP-SHA, realizando o estudo de diferentes técnicas para a identificação das sessões dos usuários que navegam *online*. Por fim, o trabalho que apresenta o método FIP-SHA obteve o aceite na conferência DEXA (NERY; GALANTE; CORDEIRO, 2021).

Conclui-se que uma das vantagens do uso do algoritmo *Affinity Propagation* é a substituição do centróide pelo exemplar, visto que são representantes dos agrupamentos e ainda são considerados dados relevantes. Apesar do algoritmo *Affinity Propagation* ser baseado em centróide e ideal para ser utilizado com a distância euclidiana, para a evolução do projeto, a avaliação utilizando-se de outros algoritmos de agrupamento e a análise da proposta com outros conjuntos de dados se faz necessária.

REFERÊNCIAS

AMATRIAIN, X. Big amp; personal: Data and models behind netflix recommendations. In: **Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications**. New York, NY, USA: Association for Computing Machinery, 2013. (BigMine '13), p. 1–6. ISBN 9781450323246. Available from Internet: <https://doi.org/10.1145/2501221.2501222>.

ARLITT, M. Characterizing web user sessions. **SIGMETRICS Perform. Eval. Rev.**, ACM, New York, NY, USA, v. 28, n. 2, p. 50–63, sep. 2000. ISSN 0163-5999. Available from Internet: <http://doi.acm.org/10.1145/362883.362920>.

BAJAJ, P.; SHEKHAR, S. Experience individualization on online tv platforms through persona-based account decomposition. In: **Proceedings of the 24th ACM International Conference on Multimedia**. New York, NY, USA: Association for Computing Machinery, 2016. (MM '16), p. 252–256. ISBN 9781450336031. Available from Internet: <https://doi.org/10.1145/2964284.2967221>.

BELL, R. M.; KOREN, Y. Lessons from the netflix prize challenge. **SiGKDD Explorations**, Citeseer, v. 9, n. 2, p. 75–79, 2007.

BOBADILLA, J. et al. Recommender systems survey. **Knowledge-Based Systems**, v. 46, p. 109 – 132, 2013. ISSN 0950-7051. Available from Internet: <http://www.sciencedirect.com/science/article/pii/S0950705113001044>.

BODENHOFER, U.; KOTHMEIER, A.; HOCHREITER, S. APCluster: an R package for affinity propagation clustering. **Bioinformatics**, v. 27, n. 17, p. 2463–2464, 07 2011. ISSN 1367-4803. Available from Internet: <https://doi.org/10.1093/bioinformatics/btr406>.

CICHOCKI, A.; PHAN, A.-H. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. **IEICE transactions on fundamentals of electronics, communications and computer sciences**, The Institute of Electronics, Information and Communication Engineers, v. 92, n. 3, p. 708–721, 2009.

DUECK, D. Affinity propagation: Clustering data by passing messages. **PhD thesis**, 01 2009.

FREY, B. J.; DUECK, D. Clustering by passing messages between data points. **Science**, American Association for the Advancement of Science, v. 315, n. 5814, p. 972–976, 2007. ISSN 0036-8075. Available from Internet: <https://science.sciencemag.org/content/315/5814/972>.

GEENG, C.; ROESNER, F. Who's in control? interactions in multi-user smart homes. In: **Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2019. (CHI '19), p. 1–13. ISBN 9781450359702. Available from Internet: <https://doi.org/10.1145/3290605.3300498>.

GOMEZ-URIBE, C. A.; HUNT, N. The netflix recommender system: Algorithms, business value, and innovation. **ACM Trans. Manage. Inf. Syst.**, ACM, New York, NY, USA, v. 6, n. 4, p. 13:1–13:19, dec. 2015. ISSN 2158-656X. Available from Internet: <http://doi.acm.org/10.1145/2843948>.

HINTON, G.; ROWEIS, S. Stochastic neighbor embedding. **Advances in neural information processing systems**, Citeseer, v. 15, p. 833–840, 2003. Available from Internet: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.7959&rep=rep1&type=pdf>.

HOTELLING, H. Analysis of a complex of statistical variables into principal components. **Journal of Educational Psychology**, American Psychological Association (APA), v. 24, n. 6, p. 417–441, 1933. Available from Internet: <https://doi.org/10.1037%2Fh0071325>.

HUANG, Y.; OBADA-OBIEH, B.; BEZNOSOV, K. K. Amazon vs. my brother: How users of shared smart speakers perceive and cope with privacy risks. In: **Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2020. (CHI '20), p. 1–13. ISBN 9781450367080. Available from Internet: <https://doi.org/10.1145/3313831.3376529>.

Jia Rongfei; Jin Maozhong; Liu Chao. A new clustering method for collaborative filtering. In: **2010 International Conference on Networking and Information Technology**. [S.l.: s.n.], 2010. p. 488–492.

JIANG, J.-Y. et al. Identifying users behind shared accounts in online streaming services. In: **The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval**. New York, NY, USA: ACM, 2018. (SIGIR '18), p. 65–74. ISBN 978-1-4503-5657-2. Available from Internet: <http://doi.acm.org/10.1145/3209978.3210054>.

JOLLIFFE, I. **Principal Component Analysis**. [S.l.]: Springer Verlag, 1986.

JOLLIFFE, I. T. Principal component analysis. In: LOVRIC, M. (Ed.). **International Encyclopedia of Statistical Science**. Springer, 2011. p. 1094–1096. ISBN 978-3-642-04898-2. Available from Internet: <http://dblp.uni-trier.de/db/reference/stat/stat2011.html#Jolliffe11>.

KAUFMAN, L.; ROUSSEEUW, P. J. **Finding Groups in Data: An Introduction to Cluster Analysis.** [S.l.]: John Wiley, 1990. ISBN 978-0-47031680-1.

LI, Y. et al. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In: **Proceedings of the 24th International Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2015. (IJCAI'15), p. 3650–3656. ISBN 9781577357384.

LIN, J. et al. "am i overwhelmed with this information?": A diary study of couples' everyday account sharing. In: **Conference Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing**. New York, NY, USA: Association for Computing Machinery, 2020. (CSCW '20 Companion), p. 311–315. ISBN 9781450380591. Available from Internet: <https://doi.org/10.1145/3406865.3418340>.

LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: Item-to-item collaborative filtering. **IEEE Internet Computing**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 7, n. 1, p. 76–80, jan. 2003. ISSN 1089-7801. Available from Internet: <http://dx.doi.org/10.1109/MIC.2003.1167344>.

LU, H.; PLATANIOTIS, K. N.; VENETSANOPOULOS, A. N. A survey of multilinear subspace learning for tensor data. **Pattern Recognition**, v. 44, n. 7, p. 1540–1551, 2011. ISSN 0031-3203. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0031320311000136>.

MA, M. et al. -net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In: **Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2019. (SIGIR'19), p. 685–694. ISBN 9781450361729. Available from Internet: <https://doi.org/10.1145/3331184.3331200>.

MAATEN, L. van der; HINTON, G. Visualizing data using t-SNE. **Journal of Machine Learning Research**, v. 9, p. 2579–2605, 2008. Available from Internet: <http://www.jmlr.org/papers/v9/vandermaaten08a.html>.

MAATEN, L. van der; HINTON, G. Visualizing data using t-sne. **Journal of Machine Learning Research**, v. 9, n. 86, p. 2579–2605, 2008. Available from Internet: <http://jmlr.org/papers/v9/vandermaaten08a.html>.

MAMMO, M.; LINDGREN, T. Evaluation of dimensionality reduction techniques-principal feature analysis in case of text classification problems. In: **Proceedings of 2020 the 6th International Conference on Computing and Data Engineering**. New York, NY, USA: Association for Computing Machinery, 2020. (ICCDE 2020), p. 75–79. ISBN 9781450376730. Available from Internet: <https://doi.org/10.1145/3379247.3379274>.

MOREIRA, G. de S. P.; FERREIRA, F.; CUNHA, A. M. da. News session-based recommendations using deep neural networks. In: **Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems**. New York, NY, USA: Association for Computing Machinery, 2018. (DLRS 2018), p. 15–23. ISBN 9781450366175. Available from Internet: <https://doi.org/10.1145/3270323.3270328>.

MOREIRA, G. de S. P.; JANNACH, D.; CUNHA, A. M. da. Contextual hybrid session-based news recommendation with recurrent neural networks. **CoRR**, abs/1904.10367, 2019. Available from Internet: <http://arxiv.org/abs/1904.10367>.

MUSA, J. M.; ZHIHONG, X. Item based collaborative filtering approach in movie recommendation system using different similarity measures. In: **Proceedings of the 2020 6th International Conference on Computer and Technology Applications**. New York, NY, USA: Association for Computing Machinery, 2020. (ICCTA '20), p. 31–34. ISBN 9781450377492. Available from Internet: <https://doi.org/10.1145/3397125.3397148>.

NEBEL, D. et al. Types of (dis-)similarities and adaptive mixtures thereof for improved classification learning. **Neurocomput.**, Elsevier Science Publishers B. V., NLD, v. 268, n. C, p. 42–54, dec. 2017. ISSN 0925-2312.

NERY, C.; GALANTE, R.; CORDEIRO, W. Fip-sha - finding individual profiles through shared accounts. In: **Int'l Conference on Database and Expert Systems Applications (DEXA2021)**. [S.l.]: Springer, 2021. (Lecture Notes in Computer Science (LNCS)), p. 1–12.

OBADA-OBIEH, B.; HUANG, Y.; BEZNOSOV, K. The burden of ending online account sharing. In: **Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2020. (CHI '20), p. 1–13. ISBN 9781450367080. Available from Internet: <https://doi.org/10.1145/3313831.3376632>.

Ponnam, L. T. et al. Movie recommender system using item based collaborative filtering technique. In: **2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)**. [S.l.: s.n.], 2016. p. 1–5.

RASTOGI, R. Machine learning @ amazon. In: **2nd IKDD Conference on Data Sciences**. New York, NY, USA: ACM, 2015. (CODS-IKDD '15), p. 2:1–2:1. ISBN 978-1-4503-3616-1. Available from Internet: <http://doi.acm.org/10.1145/2778865.2778867>.

REFIANTI, R.; MUTIARA, A.; GUNAWAN, S. Time complexity comparison between affinity propagation algorithms. **Journal of Theoretical and Applied Information Technology**, v. 95, p. 1497–1505, 04 2017.

RICCI, F. et al. **Recommender Systems Handbook**. 1st. ed. Berlin, Heidelberg: Springer-Verlag, 2010. ISBN 0387858199.

SAKIB, S.; SIDDIQUE, M. A. B.; RAHMAN, M. A. Performance evaluation of t-sne and mds dimensionality reduction techniques with knn, enn and svm classifiers. **2020 IEEE Region 10 Symposium (TENSYMP)**, IEEE, 2020. Available from Internet: <http://dx.doi.org/10.1109/TENSYMP50017.2020.9230983>.

SEMBIUM, V. et al. Bayesian models for product size recommendations. In: **Proceedings of the 2018 World Wide Web Conference**. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018. (WWW '18), p. 679–687. ISBN 9781450356398. Available from Internet: <https://doi.org/10.1145/3178876.3186149>.

SOTTOCORNOLA, G.; SYMEONIDIS, P.; ZANKER, M. Session-based news recommendations. In: **Companion Proceedings of the The Web Conference 2018**. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018. (WWW '18), p. 1395–1399. ISBN 9781450356404. Available from Internet: <https://doi.org/10.1145/3184558.3191582>.

TAN, P.-N. et al. **Introduction to Data Mining**. [S.l.]: Pearson, 2018.

UNGAR, L. H.; FOSTER, D. P. Clustering methods for collaborative filtering. In: **Workshop on Recommender Systems at the 15th National Conference on Artificial Intelligence (AAAI'98)**. Madison, Wisconsin, USA: AAAI Press, 1998. p. 112–125.

VERSTREPEN, K.; GOETHALS, B. Top-n recommendation for shared accounts. In: **Proceedings of the 9th ACM Conference on Recommender Systems**. New York, NY, USA: Association for Computing Machinery, 2015. (RecSys '15), p. 59–66. ISBN 9781450336925. Available from Internet: <https://doi.org/10.1145/2792838.2800170>.

WANG, Z. et al. User identification within a shared account: Improving ip-tv recommender performance. In: MANOLOPOULOS, Y.; TRAJCEVSKI, G.; KON-POPOVSKA, M. (Ed.). **Advances in Databases and Information Systems**. Cham: Springer International Publishing, 2014. p. 219–233. ISBN 978-3-319-10933-6.

XINHUA, H.; QIONG, W. Dynamic timeout-based a session identification algorithm. In: **2011 International Conference on Electric Information and Control Engineering**. [S.l.: s.n.], 2011. p. 346–349.

Yang, S. et al. Personalized video recommendations for shared accounts. In: **2017 IEEE International Symposium on Multimedia (ISM)**. [S.l.: s.n.], 2017. p. 256–259.

YANG, Y. et al. Adaptive temporal model for iptv recommendation. In: DONG, X. L. et al. (Ed.). **Web-Age Information Management**. Cham: Springer International Publishing, 2015. p. 260–271. ISBN 978-3-319-21042-1.

ZHANG, A. et al. Guess who rated this movie: Identifying users through subspace clustering. In: **Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence**. Arlington, Virginia, USA: AUAI Press, 2012. (UAI'12), p. 944–953. ISBN 9780974903989.