Regularização social em sistemas de recomendação com filtragem colaborativa

Tatyana Zabanova

Tatyana Zabanova

Regularização social em sistemas de recomendação com filtragem colaborativa

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP e ao Departamento de Estatística – DEs-UFSCar, como parte dos requisitos para obtenção do título de Mestra em Estatística – Interinstitucional de Pós-Graduação em Estatística. *EXEMPLAR DE DEFESA*

Área de Concentração: Estatística

Orientador: Prof. Dr. Rafael Bassi Stern

USP – São Carlos

Março de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Z12r

Zabanova, Tatyana
Regularização social em sistemas de recomendação com filtragem colaborativa / Tatyana Zabanova; orientador Rafael Bassi Stern. -- São Carlos, 2019.
67 p.

Dissertação (Mestrado - Programa Interinstitucional de Pós-graduação em Estatística) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2019.

1. Sistemas de Recomendação. 2. Filtragem Colaborativa. 3. Regularização Social. 4. Redes Sociais. I. Bassi Stern, Rafael, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa Interinstitucional de Pós-Graduação em Estatística

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado da candidata Tatyana Zabanova, realizada em 14/05/2019:

Prof. Dr. Rafael Izbicki
UFSCar

Prof. Dr. Marcelo Garcia Manzato
ICMC/USP

Prof. Dr. Marcos Oliveira Prates
UFMG

Certifico que a defesa realizou-se com a participação à distância do(s) membro(s) Marcos Oliveira Prates e, depois das arguições e deliberações realizadas, o(s) participante(s) à distância está(ao) de acordo com o conteúdo do parecer da banca examinadora redigido neste relatório de defesa.

Prof. Dr. Rafael Izbicki

Tatyana Zabanova

Social regularization in recommender systems with collaborative filtering

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP and to the Departamento de Estatística – DEs-UFSCar, in partial fulfillment of the requirements for the degree of the Master joint Graduate Program in Statistics DEs-UFSCar/ICMC-USP. *EXAMINATION BOARD PRESENTATION COPY*

Concentration Area: Statistics

Advisor: Prof. Dr. Rafael Bassi Stern

USP – São Carlos

March 2019

Dedico este trabalho ao meu gato, que com muita dedicação pediu comida e atenção nos momentos mais impróprios.

AGRADECIMENTOS

Agradeço ao meu orientador, Rafael B. Stern, pela paciência com a minha desorganização, pelos bons conselhos e, não o menos importante, pelo apoio moral nos momentos de maior estresse. Agradeço aos professores Rafael Izbicki e Hermes Senger pelas discussões, sugestões e dicas.

Agradeço aos meus pais pelo apoio, e à minha avó por perguntar toda semana se já terminei.

RESUMO

ZABANOVA, T. **Regularização social em sistemas de recomendação com filtragem colaborativa**. 2019. 81 p. Dissertação (Mestrado em Estatística – Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Modelos baseados em fatoração de matrizes estão entre as implementações mais bem sucedidas de Sistemas de Recomendação. Neste projeto, estudamos as possibilidades de incorporação de informações provindas de redes sociais, para melhorar a qualidade das predições do modelo tanto em modelos tradicionais de Filtragem Colaborativa, quanto em Filtragem Colaborativa Neural. Com base em quatro exemplos, registramos que a incorporação das informações provindas da rede social de fato leva a melhores estimativas das avaliações dadas aos itens pelos usuários.

Palavras-chave: Sistema de Recomendação, Filtragem Colaborativa, Fatoração de Matrizes, Regularização Social, Filtragem Colaborativa Neural.

ABSTRACT

ZABANOVA, T. **Social regularization in recommender systems with collaborative filtering**. 2019. 81 p. Dissertação (Mestrado em Estatística – Interinstitucional de Pós-Graduação em Estatística) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Models based on matrix factorization are among the most successful implementations of Recommender Systems. In this project, we study the possibilities of incorporating the information from social networks to improve the quality of predictions of the model both in traditional Collaborative Filtering and in Neural Collaborative Filtering. Based on four examples, we registered that incorporating information from the social network in fact leads to better estimates of the evaluations of itens by users..

Keywords: Recommender System, Collaborative Filtering, Matrix Factorization, Social Regularization, Neural Collaborative Filtering.

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

Atualmente, é oferecida ao consumidor uma imensa variedade de produtos e serviços, tornando difícil fazer uma escolha. Os sistemas de recomendação são uma ferramenta vital para auxiliar os usuários a tomar decisões com mais facilidade em um grande número de áreas, do comércio digital a redes sociais e indústria de entretenimento. Estes sistemas criam um modelo para as preferências de cada usuário (JANNACH *et al.*, 2010), de modo a recomendar uma seleção de produtos ou serviços que provavelmente será desejável, com base no comportamento dos usuários (cliques, visualizações, likes, avaliações, etc.) e nível de satisfação destes com os produtos (LU *et al.*, 2015).

Exemplos de uso bem sucedido de sistemas de recomendação podem ser encontrados em websites de comércio digital, como a Amazon, recomendações de filmes em Netflix e muitos outros. Devido ao seu potencial, sistemas de recomendação atraíram bastante atenção nas comunidades de mineração de dados (BELL; KOREN; VOLINSKY, 2007; KOREN, 2009) e machine learning (SALAKHUTDINOV; MNIH, 2007).

Uma das principais ideias ligadas ao sucesso dos Sistemas de Recomendação é a Filtragem Colaborativa (SU; KHOSHGOFTAAR, 2009), uma classe de métodos que parte dos seguintes pressupostos. Primeiro, o comportamento futuro de um consumidor pode ser previsto com base no seu comportamento no passado. Segundo, a avaliação que um consumidor faz de um item provavelmente será similar à de um item similar. Então, precisamos responder à seguinte questão: como determinar os grupos de consumidores e itens similares?

Os métodos da última década que abordaram com mais sucesso esta pergunta se baseiam em fatoração de matrizes (KOREN; BELL; VOLINSKY, 2009), método popularizado pela competição "Netflix Prize". Nestes métodos, os usuários e os itens são projetados em um mesmo sub-espaço vetorial, e a distância euclidiana é usada como medida de similaridade. Já o produto interno entre a projeção de um consumidor e a projeção de um item corresponde a uma aproximação da avaliação deste item por este consumidor. Mais recentemente, para modelar de forma mais flexível a interação entre os usuários e os itens, foi proposto o uso de Redes Neurais para Filtragem Colaborativa (HE *et al.*, 2017).

Visando melhorar a experiência do usuário, muitos sites criam redes sociais próprias. As informações provindas destas Redes Sociais, tais como informações sobre os amigos de um usuário, seu nível de confiança nos seus contatos e as atividades destes no site, podem ser usadas para melhorar o poder preditivo do Sistema de Recomendação, com o objetivo de fornecer recomendações melhores e mais personalizadas para cada usuário (MASSA; AVESANI, 2007; MA *et al.*, 2011; TANG *et al.*, 2013).

Em (MA *et al.*, 2011), especificamente, é proposto um método de recomendação com base em similaridade entre um usuário e seus amigos na rede social, que é incorporada ao processo de Filtragem Colaborativa.

Neste trabalho, o nosso objetivo é propor novas formas de incorporar informações da Rede Social do usuário ao Sistema de Recomendação, tanto em modelos da Filtragem Colaborativa clássica, quando em Filtragem Colaborativa Neural. Queremos também, com base em exemplos práticos, comparar os resultados das abordagens propostas com modelos de Filtragem Colaborativa descritos na literatura.

CAPÍTULO

2

VISÃO GERAL DAS ESTRATÉGIAS DE RECOMENDAÇÃO

Neste capítulo, vamos brevemente listar algumas estratégias de recomendação e motivar o uso de informações provindas de redes sociais para melhorar a qualidade das recomendações.

O problema de gerar uma recomendação consiste de duas partes: precisamos inferir a avaliação que um usuário daria a um determinado item e, com esta estimativa em mãos, selecionar um conjunto de itens para ser recomendado. Aqui, o foco principal é como gerar esta estimativa.

Sistemas de recomendação se baseiam em vários tipos de informação. Destes, o mais conveniente é a avaliação explícita, em que usuários informam diretamente o seu nível de satisfação com os itens. Exemplos típicos de avaliação explícita incluem likes em uma postagem nas redes sociais, e também avaliações de 0 a 5 estrelas de um determinado produto ou serviço. Contudo, este tipo de dado nem sempre está disponível, e então precisamos inferir as preferências dos usuários a partir de avaliações implícitas, muito mais abundantes. Uma avaliação implícita é aquela que não demonstra uma preferência óbvia pelo item. Exemplos destas incluem histórico de compras e padrão de buscas.

Para gerar uma recomendação a partir das informações disponíveis, existem três abordagens básicas (ADOMAVICIUS; TUZHILIN, 2005). A primeira é a Filtragem de Conteúdo (ILLIG *et al.*, 2007; BARRAGÁNS-MARTÍNEZ *et al.*, 2010), na qual se cria perfis que caracterizam todos os usuários e itens. Estes perfis permitem associar usuários a itens. A grande dificuldade desta abordagem está em montar os perfis, já que isso requer coletar informações

externas - o que pode não ser trivial.

A alternativa à Filtragem de Conteúdo se baseia somente no histórico do comportamento do usuário. Esta abordagem é conhecida como Filtragem Colaborativa (BELLOGÍN; CASTELLS; CANTADOR, 2013; CHANG; HSIAO, 2011) e é a mais popular e mais amplamente utilizada hoje em dia (G.CAMPANA; DELMASTRO, 2017). Na Filtragem Colaborativa, analisam-se as relações entre usuários e itens para identificar novas relações entre estes. A suposição básica por trás deste método é que pessoas de gostos similares terão padrões de avaliação semelhantes. As duas principais áreas da Filtragem Colaborativa são métodos de vizinhança e modelos de fator latente.

Tal como a filtragem de conteúdo, a Filtragem Colaborativa possui algumas limitações (SHAMBOUR; LU, 2012). Por exemplo, temos o problema de "cold start", quando temos um usuário ou item sem histórico de informações no sistema.

Finalmente, podemos combinar as duas abordagens (ADOMAVICIUS; TUZHILIN, 2005), de modo a minimizar os problemas e falhas inerentes a cada uma delas.

2.1 Filtragem Colaborativa

2.1.1 Métodos de vizinhança

Métodos de vizinhança, também conhecidos como métodos baseados em memória, fazem a previsão da avaliação de um item p por um usuário c com base em avaliações de p de usuários similares a c (usuário-usuário), ou com base em avaliações de itens similares a p feitas por c (item-item). A similaridade entre os usuários ou itens é calculada com base nas avaliações feitas.

A abordagem item-item mostra melhor escalabilidade e maior precisão. Além disso, ela é mais fácil de explicar e justificar ao usuário, já que este conhece os itens já avaliados, mas não conhece os tais "usuários similares".

Um método de vizinhança clássico é o kNN (k-nearest neighbors). Este método pode ser visto como uma generalização dos classificadores kNN (AGGARWAL, 2016) e consiste em medir a similaridade entre perfis de usuários do sistema para selecionar os k mais próximos ao usuário c. Os itens com alto índice de aprovação entre estes usuários mais próximos são recomendados para c. Uma ampla gama de medidas de similaridade foi proposta para ser usada em métodos de vizinhança. Destas, a Correlação de Pearson é a que parece fornecer os melhores

resultados (HERLOCKER; KONSTAN; RIEDL, 2002).

O principal problema dos métodos de vizinhança é o elevado custo computacional (G.CAMPANA; DELMASTRO, 2017), já que precisamos calcular as similaridades entre todos os pares de usuários ou de itens. Além disso, temos o problema de escalabilidade, já que para fazer uma única previsão, precisamos processar todos os dados. A esparsidade das bases de dados atuais, em que cada usuário avalia uma fração infinitesimal do conjunto de milhões de itens é um outro grande desafio, já que dificulta, ou mesmo impossibilita o cálculo da similaridade entre dois usuários.

2.1.2 Métodos de fator latente

Diferentemente dos modelos baseados em vizinhança, métodos baseados em modelos usam as avaliações observadas para treinar um modelo, e depois utilizam este modelo para prever as avaliações. Normalmente, estes modelos tentam explicar as avaliações caracterizando usuários e itens por um certo número de variáveis inferidas dos padrões de avaliação, os chamados fatores latentes. Assumimos que um número relativamente pequeno de fatores influencia as preferências do usuário, e que as preferências de cada usuário dependem de quanto cada fator se aplica a um dado usuário. Estes fatores podem ou não ter uma interpretação prática. Uma forte correspondência entre um item e um usuário resulta em recomendação.

Modelos baseados em fatoração de matrizes (KOREN; BELL; VOLINSKY, 2009) são uma das implementações mais bem sucedidas desta família de modelos, graças à sua escalabilidade e acurácia (AGGARWAL, 2016). Os problemas de escalabilidade enfrentados pelos modelos baseados em memória são em parte reduzidos, graças à capacidade do modelo de detectar as características ocultas dos dados, e consequentemente extrair mais informação (CACHEDA *et al.*, 2011). Os modelos de fatoração de matrizes também mostraram desempenho superior em várias aplicações, tais como, por exemplo, o Prêmio Netflix.

Um modelo básico de fatoração de matrizes (descrito a seguir na seção 3.2) está mostrado na Figura 1. Uma variante muito difundida deste modelo é a incorporação de viés do usuário e de item, descrito em Aggarwal (2016). No presente trabalho, usaremos esta variante (descrita na seção 3.3), como um dos modelos de referência.

Ao longo dos anos, foi proposto um grande número de diferentes modelos, incluindo Máquinas de Vetores de Suporte (XIA; DONG; XING, 2006), e, mais recentemente, Redes

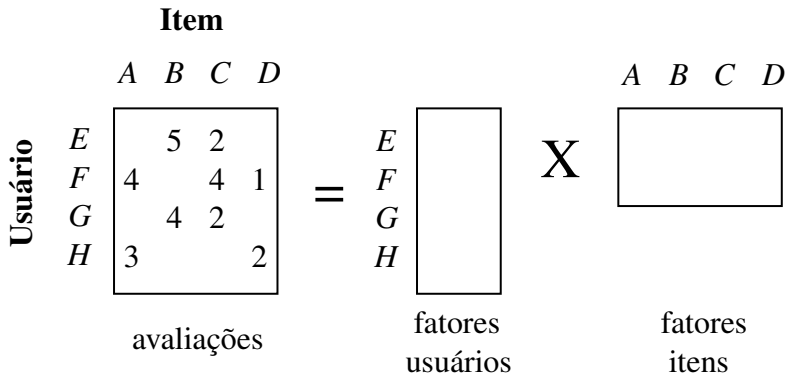

Figura 1 – Modelo básico de fatoração de matrizes

Neurais e métodos de Deep Learning (SALAKHUTDINOV; MNIH; HINTON, 2007). Em particular, em He *et al.* (2017) foi proposto um modelo de Filtragem Colaborativa Neural similar aos métodos de fator latente tradicionais. Nesta abordagem, cada usuário e item são caracterizados por embeddings. No contexto de Redes Neurais, embeddings são vetores contínuos de baixa dimensionalidade que representam variáveis discretas. Basicamente, se trata também de fatores latentes. Diferentemente dos modelos de fatoração de matrizes, em que os fatores latentes do item e do usuário são combinados via produto interno, enquanto na Filtragem colaborativa Neural, isso é feito via camadas da Rede Neural.

Os métodos de fator latente combinam facilidade de implementação, eficiência computacional e precisão relativamente alta. O ponto forte dos modelos de fatoração de matrizes é a sua flexibilidade para lidar com informações adicionais, desde avaliações implícitas até viés de avaliação.

2.2 Recomendações Sociais

Podemos definir uma "recomendação social" como "qualquer recomendação com relações sociais online como entrada adicional, por exemplo, incorporando sinais sociais adicionais a um sistema de recomendação existente" (TANG; HU; LIU, 2013).

A principal premissa desta linha de pesquisa é que as preferências de um usuário são mais similares às de seus contatos sociais do que às de usuários arbitrários. É uma hipótese razoável, já que redes sociais são uma fonte natural de recomendações: no nosso dia a dia, normalmente buscamos recomendações de pessoas que conhecemos (por exemplo, amigos ou colegas de trabalho). É cada vez mais comum uma pessoa solicitar recomendações diretamente

nas redes sociais online das quais faz parte. Além disso, o usuário pode adquirir produtos para seus contatos (por exemplo, presentes para a família), ou realizar atividades em conjunto com estes (ir a um restaurante com os amigos).

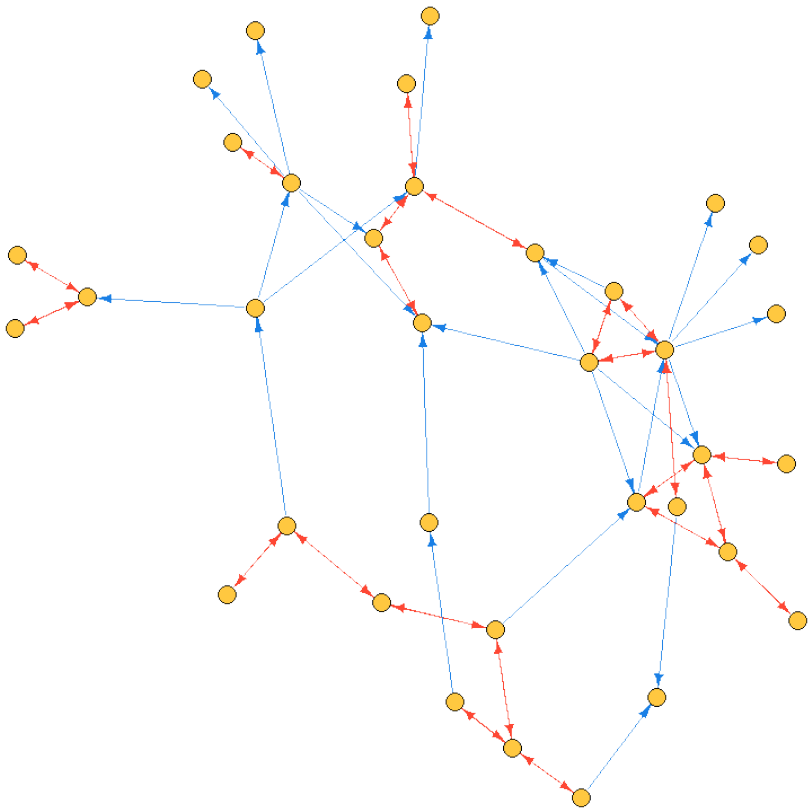

Figura 2 – Relações sociais entre usuários de uma rede social

Na Figura 2, está mostrado um grupo de pessoas de uma rede social, com as suas conexões. Recentemente, há um crescente interesse em explorar e incorporar estas informações aos sistemas de recomendação. Vários pesquisadores se dedicaram a explorar sistemas de recomendação baseados em confiança, incorporando informações de confiança do usuário em seus contatos sociais para melhorar os sistemas de recomendação tradicionais. Foram desenvolvidos vários modelos (MASSA; AVESANI, 2007; TANG *et al.*, 2013), entre os quais o proposto em Massa e Avesani (2009) é o mais popular.

Em outros estudos (PHUKSENG; SODSEE, 2017), foi abordada a possibilidade de usar a opinião de experts para melhorar o poder preditivo dos sistemas de recomendação.

Finalmente, uma vertente promissora do uso de informações de redes sociais é a incorporação de regularização social ao modelo de fator latente (MA *et al.*, 2011). Para incrementar a performance desta abordagem, foi proposto o uso de medidas de similaridade entre usuários

(MA *et al.*, 2011) e clusterização dos usuários (SUN *et al.*, 2015) para determinar aqueles cujas opiniões são mais relevantes.

CAPÍTULO

3

MÉTODO DE FATORAÇÃO DE MATRIZES

Neste capítulo, vamos definir os modelos básicos de fatoração de matrizes, tradicionalmente utilizados em sistemas de recomendação (AGGARWAL, 2016). Estes modelos servirão de base para o desenvolvimento dos modelos de fatoração de matrizes com Regularização Social.

3.1 Algumas Definições

Denotamos por V_c a lista de todos os usuários do sistema, e por V_p a lista de todos os itens. Para cada $c \in V_c$ e $p \in V_p$, $R_{c,p}$ é a avaliação numérica do usuário c para o item p.

Vamos chamar de K o conjunto de todos os pares (c, p) tais que $R_{c,p}$ é conhecido - isto é, os casos em que o usuário c deu uma avaliação para o item p. Na prática, a cardinalidade deste conjunto é muito menor do que o número de possíveis pares (c, p). O objetivo do sistema de recomendação é inferir os valores de R para os pares (c, p) que não estão contidos em K, a partir do conjunto de valores conhecidos de $R_{c,p}$.

Cada $c \in V_c$ e $p \in V_p$ é projetado em $\mathbb{R}^d$, onde d é um parâmetro do sistema de recomendação. Denotamos por C_c e P_p as projeções de, respectivamente, c e p em $\mathbb{R}^d$. Tanto os elementos de P_p, quanto os de C_c podem assumir valores positivos ou negativos.

O produto interno $C_c \cdot P_p$ modela o nível de interesse de um certo usuário c no item p. Estimamos a avaliação que o usuário c daria para o item p por:

$$\hat{R}_{c,p} = C_c \cdot P_p \,. \tag{3.1}$$

Uma vez determinados os valores de P_p e C_c, usamos a equação 3.1 para estimar a avaliação que o usuário dará a qualquer item em V_p. Caso a avaliação estimada for alta, recomendamos o item para o usuário.

3.2 Modelo A

Uma boa escolha para as projeções de c e p em $\mathbb{R}^d$, por ser a mais simples e também por ter garantias de convergência, é a que resolve o seguinte problema de otimização:

$$\underset{C_c,P_p}{\operatorname{argmin}} \sum_{(c,p)\in K} (R_{c,p} - \hat{R}_{c,p})^2$$

.

Quando R é inteiramente conhecido, C e P são obtidas projetando R nas dimensões correspondentes aos seus maiores valores singulares (ECKART; YOUNG, 1936).

Para evitar sobreajuste e forçar esparsidade, aplicamos uma penalidade, controlada pela constante λ:

$$L(C,P) = \sum_{(c,p)\in K} (R_{c,p} - C_c \cdot P_p)^2 + \lambda \left(||C_c||_F^2 + ||P_p||_F^2 \right), \tag{3.2}$$

onde a norma utilizada é a norma de Frobenius.

Para determinar o valor ideal de λ, normalmente se usa validação cruzada.

3.3 Modelo B: incorporando viés ao modelo básico

Uma fonte de variação nas avaliações é o viés. Por exemplo, alguns usuários tendem a dar uma avaliação mais positiva a todos os itens. Por outro lado, alguns itens são vistos pela população em geral como melhores (ou piores) do que os demais da mesma categoria. Por exemplo, podemos esperar que a avaliação média de um produto excelente seja mais alta do que a de um produto ruim.

Queremos estimar este viés para melhorar o poder preditivo do modelo. Podemos definir uma aproximação simples e intuitiva para o viés da seguinte forma. O viés do usuário é denotado por μ_c. Um viés de usuário positivo significa que o usuário tende a dar avaliações acima da média para todos os itens, seja por questão de gosto ou mesmo por falta de interesse em produzir uma avaliação mais criteriosa. Já um viés negativo corresponde a um usuário mais crítico. Analogamente, o viés do itens é denotado por μ_p.

Com isso, podemos quebrar a avaliação observada em quatro componentes: média geral, viés do usuário, viés do item e a interação entre usuário e item. Vamos modificar a equação 3.1 para incorporar estes viés:

$$\hat{R}_{c,p} = \mu_c + \mu_p + C_c \cdot P_p \,. \tag{3.3}$$

Vamos também incorporar o viés à equação 3.2. Definimos $L_B(C,P,\mu)$ como:

$$L_B(C,P,\mu) = \sum_{(c,p)\in K} (R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p)^2 + \lambda \left(||C_c||_F^2 + \mu_c^2 + ||P_p||_F^2 + \mu_p^2 \right) . \tag{3.4}$$

O nosso problema, assim, consiste em encontrar uma solução para o problema de otimização que minimiza $L_B(C,P,\mu)$.

Um método bastante simples e eficiente para encontrar o mínimo local neste problema é o método da descida pelo gradiente, popularizado por (FUNK, 2006). Neste procedimento, atualizamos iterativamente os parâmetros, se deslocando no sentido contrário ao gradiente. Vamos deduzir este método para o problema de minimização dado pela equação 3.4.

Definimos o erro $E_{c,p}$ como sendo:

$$E_{c,p} = R_{c,p} - (\mu_c + \mu_p + C_c \cdot P_p) \,. \tag{3.5}$$

Agora, determinamos o gradiente para as variáveis C_c, P_p, μ_c e μ_p:

$$\begin{aligned} \frac{\partial L_B(C,P,\mu)}{\partial \mu_c} &= -2 \sum_p I_{(c,p)\in K} (R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p) + 2\lambda\mu_c \\ &= -2 \left(\sum_p I_{(c,p)\in K} E_{c,p} - \lambda \mu_c \right) . \end{aligned}$$

Analogamente,

$$\frac{\partial L_B(C,P,\mu)}{\partial \mu_p} = -2 \left(\sum_c I_{(c,p)\in K} E_{c,p} - \lambda \mu_p \right) .$$

Agora, derivamos em relação a C_c:

$$\begin{aligned} \frac{\partial L_B(C,P,\mu)}{\partial C_c} &= -2 \sum_p I_{(c,p)\in K} (R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p) P_p + 2\lambda C_c \\ &= -2 \left(\sum_p I_{(c,p)\in K} E_{c,p} P_p - \lambda C_c \right) . \end{aligned}$$

E, finalmente:

$$\frac{\partial L_B(C,P,\mu)}{\partial P_p} = -2 \left(\sum_c I_{(c,p)\in K} E_{c,p} C_c - \lambda P_p \right) .$$

Agora, utilizamos o gradiente encontrado no algoritmo. Para cada combinação $(c,p) \in K$ de usuário e item, geramos uma predição $\hat{R}_{c,p}$ conforme definido na equação 3.1 e calculamos o erro associado $E_{c,p}$. Então, modificamos os parâmetros C_c e P_p proporcionalmente à taxa de aprendizado γ na direção contrária ao gradiente, obtendo:

$$\begin{aligned} \mu_c^* &\leftarrow \mu_c + \gamma(E_{c,p} - \lambda\ \mu_c)\,, \\ \mu_p^* &\leftarrow \mu_p + \gamma(E_{c,p} - \lambda\ \mu_p)\,, \\ P_p^* &\leftarrow P_p + \gamma(E_{c,p} \cdot C_c - \lambda\ P_p)\,, \\ C_c^* &\leftarrow C_c + \gamma(E_{c,p} \cdot P_p - \lambda\ C_c)\,. \end{aligned} \tag{3.6}$$

Este procedimento é repetido iterativamente até que um certo critério de parada, definido à parte, seja satisfeito. Em Funk (2006), recomenda-se utilizar $\gamma = 0{,}001$. Espera-se maior precisão ao se utilizar diferentes taxas de aprendizado γ e regularização λ para cada tipo de parâmetro. Por exemplo, é possível usar diferentes taxas de aprendizado para viés do usuário, viés do item e os próprios fatores.

3.4 Modelos C e D: norma L^1

Tradicionalmente, modelos de fatoração de matrizes utilizam norma de Frobenius. Contudo, em outras áreas, a norma L^1 é largamente utilizada em penalizações (por exemplo, Lasso), por ter, também, uma derivada de cálculo simples. Assim, propomos estender esta abordagem também para os modelos de fatoração de matrizes e trabalhar com a norma L^1. Assim, usando o $\hat{R}_{c,p}$ dado pela Equação 3.5, definimos o Modelo C:

$$L_C(C,P,\mu) = \sum_{(c,p)\in K} |R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p| + \lambda G(C_c,P_p,\mu)\,. \tag{3.7}$$

Analogamente, o Modelo D:

$$L_D(C,P,\mu) = \sum_{(c,p)\in K} (R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p)^2 + \lambda G(C_c,P_p,\mu)\,. \tag{3.8}$$

Nas duas equações acima, G é dado por:

$$G(C_c,P_p,\mu) = |C_c|_1 + |\mu_c|_1 + |P_p|_1 + |\mu_p|_1\,.$$

A derivação detalhada dos gradientes para estes dois modelos pode ser encontrada no apêndice A.1.

CAPÍTULO

4

INCORPORANDO REDES SOCIAIS NO MODELO TÍPICO DE SISTEMAS DE RECOMENDAÇÃO

Neste capítulo, vamos apresentar alguns modelos de incorporação de informações da Rede Social ao Sistema de Recomendação sugeridos na literatura, e principalmente sugerir algumas abordagens novas.

Vamos começar com algumas definições referentes a redes sociais. Definimos a matriz de relações sociais $M_{|V_c|\times|V_c|}$ tal que:

$$M_{c_1,c_2} = \begin{cases} 1\,, & \text{se } c_1 \text{ é amigo de } c_2 \\ 0\,, & \text{c.c.} \end{cases}$$

A matriz M será simétrica em caso de relações sociais mútuas.

Definimos a lista de amigos do usuário c_1 como sendo o conjunto de usuários que são amigos de c_1:

$$\mathbb{F}(c_1) = \{c_2 \in V_c : c_1 \neq c_2 \cap M_{c_1,c_2} = 1\}\,.$$

Analogamente, a lista de usuários de quem c_1 é amigo é dada por:

$$\mathbb{F}^-(c_1) = \{c_2 \in V_c : c_1 \neq c_2 \cap M_{c_2,c_1} = 1\}\,.$$

Se as relações sociais forem direcionadas, estas duas listas podem ser diferentes. Já se as relações sociais forem mútuas, as duas listas $\mathbb{F}(c_1)$ e $\mathbb{F}^-(c_1)$ coincidem.

O nosso problema é diferente dos sistemas de recomendação tradicionais, pois estes consideram somente a matriz de usuários - itens. Neste capítulo, vamos focar em incorporar as informações de interações da rede social do usuário aos modelos com viés (Modelos B, C e D) descritos no capítulo anterior.

Como o usuário c é descrito por um fator C_c e por um viés μ_c, podemos abordar este problema de duas formas. A primeira é fazer regularização social para o fator do usuário, enquanto a segunda é aplicar a regularização ao viés. Podemos também combinar as duas abordagens.

4.1 Regularização Social para o Fator do Usuário

4.1.1 Modelo 1: distância ao centro

Os dois modelos apresentados nesta seção foram propostos em (MA *et al.*, 2011). A primeira ideia de como introduzir informações de redes sociais no modelo é penalizar a distância de cada usuário até o centro dos usuários que são seus amigos. No primeiro modelo, $L_1^*(C,P,\mu)$, a penalização se baseia na suposição de que o gosto do usuário c é similar à média dos gostos dos seus amigos.

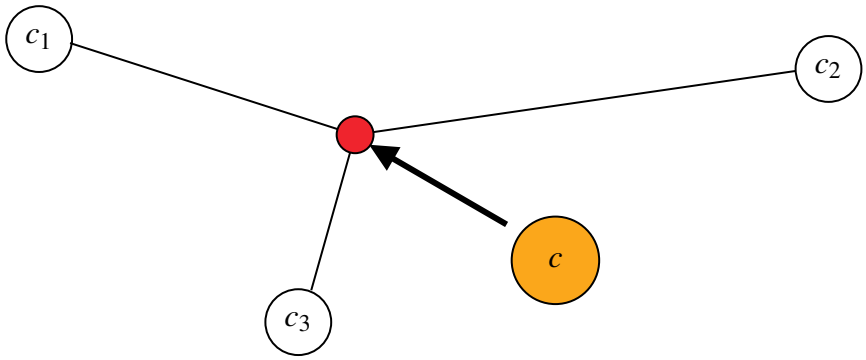

Figura 3 – Distância ao centro

A Figura 3 mostra a ideia básica deste modelo. Temos um usuário c, com três amigos, c_1, c_2 e c_3. Esperamos que os gostos (ou, em termos do modelo, o fator latente) de c sejam próximos a média dos gostos (em vermelho) dos seus amigos. Queremos definir a penalização de tal forma que ela force o fator C_c a se aproximar desta média.

Seja $\alpha > 0$. Definimos o Modelo 1 como sendo:

$$L_1^*(C,P,\mu) = L_B(C,P,\mu) + \alpha \sum_{c \in V_c} \left|\left| C_c - \frac{\sum_{c_2 \in \mathbb{F}(c)} C_{c_2}}{|\mathbb{F}(c)|} \right|\right|_F^2 . \tag{4.1}$$

Acima, o centro de um grupo é definido pela média usual. Contudo, a suposição sobre a proximidade dos gostos de um usuário e os gostos dos seus amigos não é necessariamente verdadeira.

De modo geral, é razoável supor que amigos têm, de fato, gostos mais similares do que a média. Mas um usuário pode também ter amigos ou outros contatos sociais com gostos e interesses distintos dos seus. Assim, um determinado usuário do Facebook tem, entre seus "amigos" nesta rede social, amigos próximos que compartilham os seus interesses, familiares e colegas de trabalho que foram adicionados à rede por motivos que não são similaridade de gostos, e contatos casuais que podem muito bem ser uma amostra aleatória da população.

Para contemplar essa semelhança (ou não) entre o usuário e seus amigos, vamos utilizar uma medida de similaridade genérica, que assume valores próximos de 1 quando dois usuários são muito parecidos, e próximos de 0 quando são muito diferentes. Nas próximas seções, discutiremos várias medidas de similaridade em mais detalhes.

Vamos incorporar a similaridade ao Modelo 1:

$$L_{1S}^*(C,P,\mu) = L_B(C,P,\mu) + \alpha \sum_{c \in V_c} \left|\left| C_c - \frac{\sum_{c_2 \in \mathbb{F}(c)} Sim(c,c_2) C_{c_2}}{\sum_{c_2 \in \mathbb{F}(c)} Sim(c,c_2)} \right|\right|_F^2 . \tag{4.2}$$

Neste modelo, definimos o centro de um grupo pela média ponderada pela similaridade. Por exemplo, se dois consumidores c_1 e c_2 são parecidos - ou seja, $Sim(c_1,c_2)$ é próxima de 1 - as opiniões do usuário c_2 terão maior peso do que a média ao se recomendar itens para c_1.

Observe que o Modelo 1 com similaridade, se $Sim(c_i,c_j)$ for igual a 1 para todos os pares de usuários, é equivalente ao Modelo 1 sem similaridade.

Os gradientes para estes dois modelos podem ser encontrados no apêndice A.2.

4.1.2 Modelo 2: propagação de gostos

O Modelo 1 apresenta uma falha óbvia: é pouco sensível quando se trata de um usuário cujos amigos têm gostos muito variados. Para contornar este problema, (MA *et al.*, 2011) propõe

mais uma forma de regularização, que vamos chamar de Modelo 2:

$$L_2^*(C,P,\mu) = L_B(C,P,\mu) + \alpha \sum_{c \in V_c} \sum_{c_2 \in \mathbb{F}(c)} ||C_c - C_{c_2}||_F^2 . \tag{4.3}$$

Uma vantagem desta abordagem é que ela modela diretamente a propagação dos gostos e interesses. Suponhamos que, na Figura 4, o usuário c_1 seja amigo de c_2, e c_2 seja amigo de c_3, mas c_1 e c_3 não sejam amigos. Ao minimizar $d_1 = ||C_{c_1} - C_{c_2}||_F^2$ e $d_2 = ||C_{c_2} - C_{c_3}||_F^2$, minimizamos indiretamente a distância d_3 entre os vetores C_{c_1} e C_{c_3}.

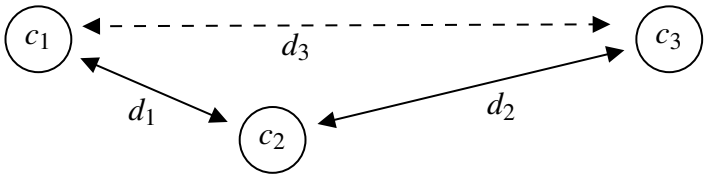

Figura 4 – Propagação de gostos

Seguindo a mesma lógica apresentada no Modelo 1, podemos ponderar as normas das diferenças pela similaridade entre dois usuários:

$$L_{2S}^*(C,P,\mu) = L_B(C,P,\mu) + \alpha \sum_{c \in V_c} \sum_{c_2 \in \mathbb{F}(c)} Sim(c,c_2)||C_c - C_{c_2}||_F^2 . \tag{4.4}$$

O gradiente para este modelo pode ser encontrado no apêndice A.3

4.1.3 Modelo 3: propagação de gostos com norma L^1

Ao invés de utilizar a norma de Frobenius, tal como é feito em Ma *et al.* (2011), propomos utilizar a norma L^1 para a regularização social.

$$L_3^*(C,P,\mu) = L_{C/D}(C,P,\mu) + \alpha \sum_{c \in V_c} \sum_{c_2 \in \mathbb{F}(c)} ||C_c - C_{c_2}||_1 . \tag{4.5}$$

A regularização será aplicada ao modelo C ou D, com base na análise da performance destes dois modelos.

Seguindo a mesma lógica aplicada aos Modelos 1 e 2, podemos ponderar as normas das diferenças pela similaridade entre dois usuários.

$$L_{3S}^*(C,P,\mu) = L_{C/D}(C,P,\mu) + \alpha \sum_{c \in V_c} \sum_{c_2 \in \mathbb{F}(c)} Sim(c,c_2)||C_c - C_{c_2}||_1 . \tag{4.6}$$

O gradiente para este modelo pode ser encontrado no apêndice A.4

4.1.4 Medidas de Similaridade

Como já discutimos anteriormente, um usuário pode ter tanto amigos com gostos parecidos com os seus, quanto amigos com gostos diferentes dos seus. Portanto, queremos uma medida de similaridade que permita quantificar estas diferenças.

Uma ideia para identificar o quão similares são dois usuários é analisar as suas avaliações dos itens. Afinal, dois usuários que compartilham os mesmos interesses e têm as mesmas características darão avaliações similares aos mesmos itens. Da mesma forma, é razoável supor que o inverso é válido: se dois usuários avaliam da mesma forma um conjunto de itens, é provável que tenham características similares.

Assim, com base no conjunto K de avaliações $R_{c_1,p}$ e $R_{c_2,p}$ conhecidas, queremos definir uma função de similaridade entre os usuários c_1 e c_2.

Definimos o conjunto de itens avaliados pelos usuários c_1 e c_2 como $I(c_1) \cap I(c_2)$, onde $I(c_1) = \{p : (c_1, p) \in K\}$ e $I(c_2) = \{p : (c_2, p) \in K\}$ são os conjuntos de itens avaliados por cada um dos usuários.

Em Ma *et al.* (2011), são propostas duas medidas de similaridade.

A primeira, Similaridade de Espaços Vetoriais (SEV), é dada por:

$$Sim_1(c_1, c_2) = \frac{\sum_{p \in I(c_1) \cap I(c_2)} R_{c_1,p} \cdot R_{c_2,p}}{\sqrt{\sum_{p \in I(c_1) \cap I(c_2)} R_{c_1,p}^2} \cdot \sqrt{\sum_{p \in I(c_1) \cap I(c_2)} R_{c_2,p}^2}}, . \tag{4.7}$$

Observamos que $Sim(c_1, c_2) \in [0, 1]$. Quanto mais próximo de 1 o valor de $Sim(c_1, c_2)$, maior a similaridade entre usuários. Já 0 corresponde a usuários inteiramente diferentes. Quando dois usuários avaliaram somente um item em comum, temos que $Sim_1(c_1, c_2) = 1$, por definição.

A segunda função proposta em (MA *et al.*, 2011) é o Coeficiente de Correlação de Pearson (CCP), função de similaridade que é também usada em métodos de vizinhança de Filtragem Colaborativa, com bons resultados (G.CAMPANA; DELMASTRO, 2017):

$$Sim_2(c_1, c_2) = \frac{\sum_{p \in I(c_1) \cap I(c_2)} (R_{c_1,p} - \bar{R_{c_1}}) \cdot (R_{c_2,p} - \bar{R_{c_2}})}{\sqrt{\sum_{p \in I(c_1) \cap I(c_2)} (R_{c_1,p} - \bar{R_{c_1}})^2} \cdot \sqrt{\sum_{p \in I(c_1) \cap I(c_2)} (R_{c_2,p} - \bar{R_{c_2}})^2}}, \tag{4.8}$$

onde $\bar{R_c}$ é a média das avaliações do usuário c. Neste caso, $Sim_2(c_1, c_2) \in [-1, 1]$. Novamente, quanto maior o valor de $Sim_2(c_1, c_2)$, maior a similaridade entre usuários. Para manter a consistência entre as medidas, vamos mapear esta medida de similaridade para o intervalo $[0, 1]$,

aplicando a função $f(x) = (x+1)/2$ ao CCP, conforme sugerido em (MA *et al.*, 2011). Observe que, quando dois usuários avaliam um único item em comum, $Sim_2(c_1, c_2) = 0$, quaisquer que sejam as avaliações.

Além disso, neste trabalho, propomos outras quatro medidas de similaridade.

A medida de similaridade a seguir é baseada em seguintes suposições. O simples fato de que um certo usuário avalia um item indica o interesse do usuário em itens deste tipo. Por exemplo, temos um usuário que gosta de comédias, mas não gosta de filmes de terror. Este usuário irá assistir filmes dos quais potencialmente pode gostar, que são as comédias, podendo portanto avaliá-los, positivamente ou negativamente, mas não vai assistir filmes de terror e, consequentemente, não poderá avaliar estes filmes.

Dessa forma, propomos usar o Índice de Jaccard (JACCARD, 1912) para medir a similaridade entre usuários. Este índice é dado por:

$$Sim_3(c_1, c_2) = \frac{|I(c_1) \cap I(c_2)|}{|I(c_1) \cup I(c_2)|}. \tag{4.9}$$

Finalmente, podemos usar como medida de similaridade a norma da diferença entre as avaliações:

$$S(c_1, c_2) = \frac{\sum\limits_{p \in I(c_1) \cap I(c_2)} |R_{c_1,p} - R_{c_2,p}|}{|I(c_1) \cap I(c_2)|}. \tag{4.10}$$

Para restringir esta medida ao intervalo $[0, 1]$, adotamos as duas transformações abaixo:

$$Sim_4(c_1, c_2) = \frac{1}{1 + S(c_1, c_2)}, \tag{4.11}$$

$$Sim_5(c_1, c_2) = e^{-S(c_1,c_2)}. \tag{4.12}$$

Finalmente, podemos pensar em uma similaridade exponencial mais genérica, adicionando um parâmetro β à Sim_5:

$$Sim_6(c_1, c_2) = e^{-\beta S(c_1,c_2)}, \tag{4.13}$$

onde $S(c_1, c_2)$ é dado pela equação 4.10. O β pode ser determinado pela validação cruzada, ou o seu cálculo incorporado na descida pelo gradiente.

Para o Modelo 1, a derivada da função de perda em β tem uma forma complexa, o que resulta em maior custo computacional. Por isso, não vamos estudar esta combinação de modelo e função de similaridade neste trabalho.

Se simplesmente substituirmos Sim_6 nas equações dos Modelos 2 e 3, o valor de β que minimiza a função resultante será aquele que resulta em valores menores de similaridade. Isto é, teremos a solução trivial $\beta = \infty$. Para evitar que isso aconteça, vamos incluir uma penalização adicional, usando a seguinte função de perda para o Modelo 2:

$$L^*_{2S\beta}(C,P,\mu) = L_B(C,P,\mu) + \alpha \left(\sum_{c \in V_c} \sum_{c_2 \in \mathbb{F}(c)} e^{-\beta S(c,c_2)} ||C_c - C_{c_2}||^2_F + \beta^2 \right). \tag{4.14}$$

Mais detalhes sobre o gradiente podem ser encontrados no apêndice A.5.

4.1.5 Possíveis problemas com similaridade

Todas as funções de similaridade apresentadas acima, com exceção de Sim_3, se baseiam em itens avaliados tanto pelo usuário, quanto pelo seu amigo. Devido a elevada esparsidade das bases de dados com quais tipicamente lidamos em problemas de recomendações sociais, é possível que a grande maioria dos pares de amigos não tenha itens avaliados em comum.

Uma solução trivial, neste caso, é utilizar os amigos do amigo de um usuário para o cálculo da similaridade. Por exemplo, se $I_{am}(c) = \bigcup_{i \in F(c)} I(i)$ o conjunto de itens avaliados por amigos de um usuário e $A(c_1,c_2) = I(c_1) \cap I_{am}(c_2)$ o conjunto de itens avaliados tanto pelo usuário c_1 quanto por amigos do usuário c_2, definimos a avaliação média do item p por amigos do usuário c:

$$\overline{R}(c,p) = \frac{\sum_{i \in \{i \in F(c_2) : p \in I(c)\}} R_{c,p}}{|\{i \in F(c_2) : p \in I(c)\}|}.$$

Com isso, podemos re-escrever a similaridade $Sim_1(c_1,c_2)$ como:

$$Sim_1(c_1,c_2) = \frac{\sum_{p \in A(c_1,c_2)} R_{c_1,p} \cdot \overline{R}_{c_2,p}}{\sqrt{\sum_{p \in A(c_1,c_2)} R^2_{c_1,p}} \cdot \sqrt{\sum_{p \in A(c_1,c_2)} \overline{R}^2_{c_2,p}}}. \tag{4.15}$$

Analogamente, podemos usar $\overline{R}(c,p)$ nas demais similaridades.

4.2 Regularização Social para o Viés do Usuário

Podemos supor, também, que dois usuários amigos tendem a ter um viés semelhante. Por exemplo, ambos vão dar avaliações acima ou abaixo da média. Isso parece menos plausível do que a semelhança dos fatores de usuário, mas ainda assim é uma hipótese que merece ser testada.

Com base neste pressuposto, propomos o seguinte modelo com regularização social:

$$L_V^*(C,P,\mu) = L(C,P,\mu) + \alpha \sum_{c \in V_c} \sum_{c_2 \in \mathbb{F}(c)} |\mu_c - \mu_{c_2}|^2 . \tag{4.16}$$

Chamaremos este modelo de Modelo 4.

Conforme discutimos anteriormente, é possível que nem todos os amigos de um usuário compartilhem a sua visão positiva (ou negativa) dos itens de modo geral. Assim, queremos também um modelo que contemple estas possíveis diferenças:

$$L_{VS}^*(C,P,\mu) = L(C,P,\mu) + \alpha \sum_{c \in V_c} \sum_{c_2 \in \mathbb{F}(c)} Sim(c,c_2)|\mu_c - \mu_{c_2}|^2 . \tag{4.17}$$

Estas duas regularizações podem ser facilmente incorporadas aos Modelos 1, 2 e 3, em adição à regularização social para fatores dos usuários.

CAPÍTULO

5

FILTRAGEM COLABORATIVA NEURAL

Uma grande limitação dos modelos anteriores é o uso do produto interno para descrever a relação entre os fatores latentes do usuário e do item. Nos capítulos anteriores, para lidar com este problema, incorporamos viés de usuário e de item ao modelo. Uma outra solução possível é a Filtragem Colaborativa Neural, descrita em He *et al.* (2017) , cujo objetivo é exatamente captar a complexidade da relação entre usuários e itens.

Inicialmente, este enfoque não fazia parte do projeto, contudo a abordagem se mostrou bastante promissora, por isso optamos por explorar alguns exemplos básicos, para mostrar seu imenso potencial.

5.1 Modelo básico: MLP

Neste modelo, proposto em He *et al.* (2017), modelamos a avaliação $R_{c,p}$ do item p pelo usuário c conforme mostrado na Figura 5.

A entrada consiste em identificação do usuário e do item, com codificação one-hot. Acima desta, é a camada de embedding, que projeta a representação esparsa para um vetor denso. O embedding para um certo usuário ou item pode ser interpretado como um fator latente.

Finalmente, os embeddings do usuário e do item são passados para o conjunto de camadas da filtragem colaborativa neural, com X camadas no total. O número e as características destas camadas podem ser personalizados em função do problema de recomendação.

A forma mais intuitiva de combinar os dois embeddings, e muito usada em deep learning

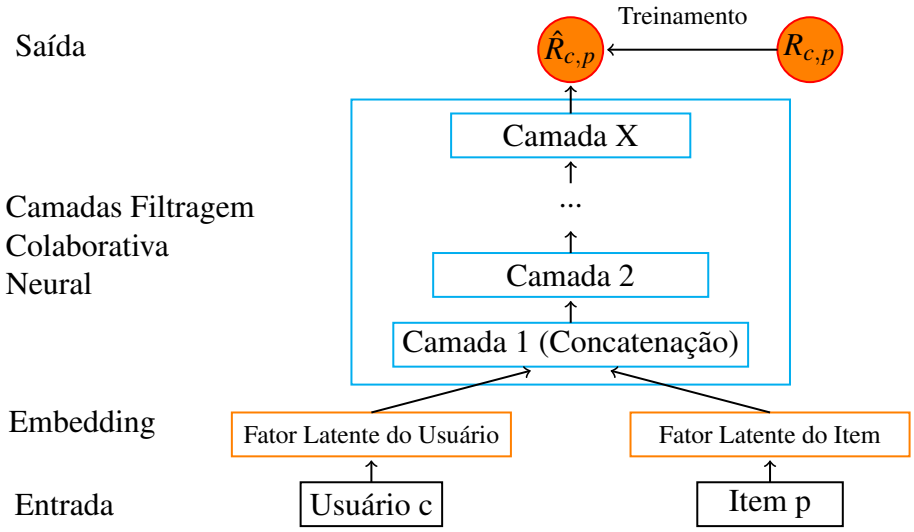

Figura 5 – Filtragem Colaborativa Neural

(por, exemplo, em Srivastava e Salakhutdinov (2012)), é a concatenação. Mas esta, por si só, não é suficiente, pois não modela as interações entre usuários e itens. Por isso, conforme proposto em He *et al.* (2017), vamos usar Perceptron de Múltiplas Camadas. Esta estrutura permite grande flexibilidade no aprendizado das interações entre usuários e itens, incluindo interações não lineares, e nisso difere do produto interno usado nos capítulos anteriores.

No contexto de Filtragem Colaborativa Neural, definimos o modelo de Perceptron de Múltiplas Camadas da seguinte forma:

$$\begin{aligned} z_1 &= \phi_1(C_c, P_p) = \begin{bmatrix} C_c, \\ P_p \end{bmatrix}, \\ \phi_2(z_1) &= a_2(W_2^T z_1 + b_2), \\ &\dots \\ \phi_x(z_{x-1}) &= a_x(W_x^T z_{x-1} + b_x), \\ \hat{R}_{c,p} &= \sigma(h^T \phi_x(z_{x-1})), \end{aligned} \tag{5.1}$$

onde W_i, b_i e a_i denotam, respectivamente, pesos, viés e função de ativação da i-ésima camada do Perceptron de Múltiplas Camadas, e C_c e P_p são os fatores latentes do usuário e do item.

A camada de saída corresponde à avaliação estimada $\hat{R}_{c,p}$. Treinamos o modelo minimizando a perda em relação à avaliação real $R_{c,p}$. A função de perda é dada por:

$$\sum_{(c,p)\in K} (R_{c,p} - \hat{R}_{c,p})^2, \tag{5.2}$$

onde K é o conjunto de todos os pares (c,p) tais que $R_{c,p}$ é conhecido, definido em 3.1.

Dessa forma, temos o modelo básico, ao qual queremos incorporar as informações provenientes da rede social.

5.2 Incorporando informações da Rede Social

Devido a grande flexibilidade das redes neurais, existem inúmeras formas de incluir as informações provenientes das redes sociais na Filtragem Colaborativa Neural. Uma das mais óbvias é resumir estes dados em um número ou vetor, e concatenar junto com os fatores latentes do usuário e do item, conforme mostrado na Figura 6.

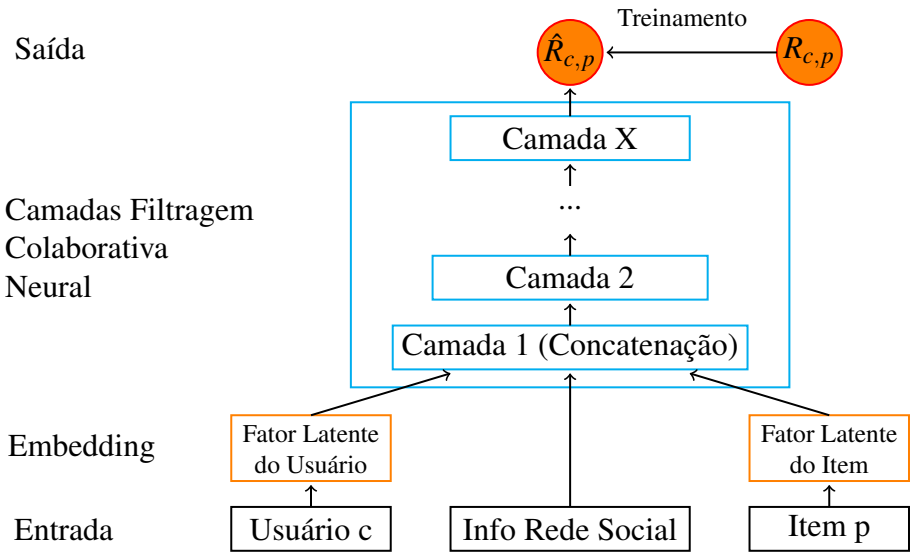

Figura 6 – Filtragem Colaborativa Neural com uso de informações da Rede Social

Na equação 5.1, teríamos, então:

$$z_1 = \phi_1(C_c, P_p) = \begin{bmatrix} C_c \\ P_p \\ S_c \end{bmatrix}, \tag{5.3}$$

onde S_c é alguma informação proveniente dos amigos do usuário c, o restante do modelo se mantendo inalterado. Assim, precisamos escolher formas de resumir os dados sobre os amigos de um determinado usuário e as suas preferências.

Vamos propor três modelos baseados no modelo de Filtragem Colaborativa Neural.

5.2.1 Modelo 5

Se queremos estimar o interesse do usuário c em um determinado item p com base nas preferências de seus amigos, a média das avaliações deste item pelos amigos do usuário parece um estimador natural. Dessa forma, definimos:

$$S_1(c,p) = \frac{\sum\limits_{(c_2 \in \mathbb{F}(c)) \cap (p \in I(c_2))} R_{c_2,p}}{|(c_2 \in \mathbb{F}(c)) \cap (p \in I(c_2))|}. \tag{5.4}$$

Um problema com esta abordagem é a esparsidade, já que é possível que, para um usuário c e item p, nenhum dos amigos de c tenha avaliado este item. Neste caso, vamos usar a avaliação média global de p como S_c, excluindo-se a avaliação dada pelo próprio usuário. Caso o item tenha sido avaliado somente por um usuário, tomamos $S_c = 0$.

5.2.2 Modelo 6

As avaliações dadas aos itens são discretas. Assim, podemos pensar nas avaliações dadas pelos amigos de um certo usuário como identidades com seus próprios fatores latentes, que de alguma forma contribuem para a avaliação dada pelo usuário.

Assim segunda possibilidade é incorporar ao modelo também um "fator latente" da avaliação dada pelos amigos de um usuário. Vamos usar a mediana, uma forma natural de resumir as avaliações dos amigos de um usuário em um único valor com a mesma granularidade das avaliações originais:

$$S_2(c,p) = \underset{(c_2 \in \mathbb{F}(c)) \cap (p \in I(c_2))}{\text{mediana}} \{R_{c_2,p}\}. \tag{5.5}$$

Tal como no modelo anterior, usamos a mediana global caso não haja avaliações em comum, e usamos 0 caso o item tenha sido avaliado somente uma vez.

Aplicamos codificação one-hot e adicionamos um terceiro embedding, que vai ser concatenado na Camada 1 junto com os embeddings do usuário e do item. Esta estrutura está mostrada na Figura 7.

5.2.3 Modelo 7

Finalmente, podemos usar $M_{|V_c| \times |V_c|}$, a matriz de relações sociais dada pela equação 4. Neste caso, a nossa estatística não depende das avaliações dadas, mas somente das relações de

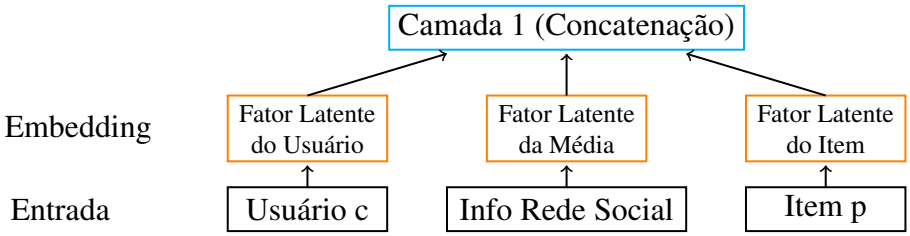

Figura 7 – Embedding para a informação da rede social

amizade. Certamente, poderíamos definir uma matriz $M(p)$, contendo também as avaliações, mas teríamos que enfrentar novamente o problema da esparsidade, e do elevado número de usuários cujos amigos avaliaram itens inteiramente diferentes.

Assim, vamos simplesmente tomar S_c como sendo a c-ésima linha desta matriz:

$$S_3(c) = M_{c,*}\,. \tag{5.6}$$

5.3 Outras Possibilidades

Neste ponto, convém levantar a pergunta lógica: porque não incorporamos as informações da Rede Social usando algum procedimento similar ao que foi feito para modelos de Fatoração de Matrizes? Como trabalhamos com fatores latentes do usuário em ambos os modelos, poderíamos incorporar a informação da rede social na Filtragem Colaborativa Neural usando o próprio embedding do usuário, em uma estrutura similar à mostrada na Figura 8.

A grande restrição dessa abordagem é que o modelo MLP não tem uma forma trivial de usar entradas de tamanho variado e o número de amigos justamente varia de usuário para usuário,

Uma primeira ideia seria, no treinamento, a cada iteração selecionar um amigo ao acaso para cada usuário, e usar uma estrutura similar à mostrada abaixo. Contudo, enquanto isso parece factível no treinamento, a questão de como produzir estimativas com este modelo precisaria de mais investigação. Selecionaríamos também um amigo ao acaso? Ajustaríamos o modelo para cada um dos amigos, e combinaríamos os ajustes usando média ou mediana?

Uma outra possibilidade é converter a entrada de tamanho variável em uma de tamanho fixo usando a técnica de “zero padding”. Basicamente, definimos uma entrada de tamanho n suficientemente grande, e completamos com zeros se o tamanho de uma entrada específica - isto é, o número de amigos, for menor do que n. Em suma, completamos o conjunto de amigos do

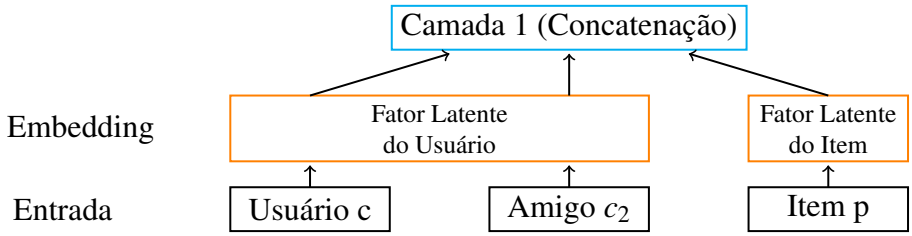

Figura 8 – Embedding do usuário compartilhado entre o amigo e o usuário

usuário com amigos “dummy”.

Finalmente, há a possibilidade de se afastar da arquitetura de MLP, algo que não abordamos para não fugir do foco do trabalho.

CAPÍTULO

6

ANÁLISE EXPERIMENTAL

Os modelos de fatoração de matrizes, descritos no capítulo 4, foram implementados usando-se a linguagem de programação Python. Além disso, para fins de comparação, foram também implementados os modelos sem Regularização Social (Modelos B, C e D). Os modelos de Filtragem Colaborativa Neural também foram implementados em python, com uso do pacote Keras.

Os detalhes da implementação do Gradiente Descendente Estocástico podem ser encontrados no apêndice B.

Neste capítulo, vamos apresentar os resultados obtidos na aplicação destes modelos em quatro bases de dados reais.

6.1 Métrica Utilizada

Utilizamos, como métrica para avaliar a qualidade do ajuste, o Erro Quadrático Médio (EQM), calculado em uma base de teste. Seja T o número de avaliações testadas, $R_{c,p}$ a avaliação dada pelo usuário c para o item p, e $\hat{R}_{c,p}$ a estimativa obtida, então:

$$EQM = \frac{1}{T}\sum_{c,p}(R_{c,p} - \hat{R}_{c,p})^2 . \tag{6.1}$$

Quanto menor o valor de EQM, melhor a performance do modelo.

6.2 Critério de Parada

Nos experimentos iniciais, observou-se que o critério de parada dos algoritmos baseado em convergência, isto é, parada quando a variação de um indicador de um passo para o seguinte fica abaixo de um determinado threshold, leva ao sobreajuste e consequente piora do desempenho. Uma solução para isso são os critérios de parada precoce.

Trabalharemos, assim, não só com bases de Treinamento e de Teste, mas também de Validação. Seja $EQM_{tr}(t)$ o EQM na base de Treinamento na t-ésima iteração e EQM_{val} o EQM na base de Validação, definimos o EQM Ótimo como sendo $EQM_{opt} = \min_{T<t} EQM_{val}(T)$. A ideia básica é detectar que o EQM_{val} já passou pelo ponto de mínimo, e tomar por solução o ajuste correspondente ao EQM_{opt} neste intervalo. Isso parece simples na teoria, mas na prática as curves de EQM_{val} frequentemente possuem vários mínimos locais, dificultando a tarefa.

Um critério parada descrito na literatura (PRECHELT, 1998) é o GL_α. Vamos definir a perda generalizada na iteração t como sendo o aumento relativo em relação ao EQM_{opt} (até a iteração t):

$$GL(t) = \frac{EQM_{val}(t)}{EQM_{opt}(t)} - 1\,. \tag{6.2}$$

Paramos a execução do algoritmo quando $GL(t)$ atinge um limiar pré-definido. Usaremos o valor de $0,02$ que mostrou bom desempenho tanto no estudo em questão, quanto em testes iniciais com as nossas bases de dados. Caso o critério de parada não seja satisfeito, paramos o ajuste após 1.000 iterações.

Conforme mostrado em Prechelt (1998), esta família de critérios mostra bom desempenho quando estamos interessados em encontrar uma "boa" solução. Assim, o nosso procedimento é usar uma base de Treinamento (60% da base de dados) para estimar os parâmetros de cada modelo, uma base de Validação (20%) para estabelecer o ponto de parada do algoritmo, e, finalmente, uma base de Teste (20%) para comparar o desempenho de modelos diferentes.

Estas bases serão sorteadas ao acaso. Certamente, o ideal seria seria separar as avaliações mais recentes para usar como Teste e Validação, contudo a informação sobre a data e hora da avaliação não estava disponível em todas as bases, portanto usamos por adotar uma abordagem padrão - sorteio aleatório - para as quatro.

6.3 Escolha dos parâmetros

6.3.1 Parâmetros dos modelos de Fatoração de Matrizes

Cada um dos modelos de Fatoração de Matrizes possui três parâmetros, λ, α e número de dimensões d. Levando em conta que temos um total de 24 modelos a ajustar para cada uma das bases, fazer validação cruzada em três dimensões para cada ajuste seria inviável. Assim, optamos por usar $\lambda = \alpha$, e utilizar mesmos valores para todos os ajustes.

Os valores utilizados, para cada base, foram encontrados fazendo-se validação cruzada para o Modelo B.

Usamos a taxa de aprendizado $\gamma = 0,001$, seguindo a recomendação de Funk (2006). Experimentamos também usar outros valores, mas de modo geral se afastar muito deste valor de γ ou levou à instabilidade, ou resultou em elevado tempo de execução.

6.3.2 Arquitetura dos modelos de Filtragem Colaborativa Neural

Para os modelos de Filtragem Colaborativa Neural, o nosso foco é simplesmente incorporar as informações sobre amigos ao modelo. Experimentamos também com modelos mais complexos, incluindo múltiplas camadas e diversas funções de ativação, contudo isso não levou a uma melhora radical nos resultados do ajuste. Assim, para evitar a introdução de arquiteturas cujo uso dificilmente poderíamos justificar, optamos por trabalhar com a estrutura mais simples possível, com uma única camada densa.

A dimensão dos embeddings foi fixada, arbitrariamente, em 20. Experimentamos com uma variedade de valores para este parâmetro, contudo, para as bases de dados testadas, o impacto não foi significativo.

6.4 Bases de dados utilizadas

Nesta seção, listamos as bases de dados usadas no projeto, e algumas estatísticas básicas. Informações mais detalhadas podem ser encontradas nos apêndices.

6.4.1 Epinions

Epinions.com foi um site contendo reviews de produtos por usuários que funcionou entre 1999 e 2014. Neste site, usuários poderiam ler e escrever avaliações sobre uma grande variedade de produtos, e também indicar usuários em cujas opiniões confiam. Epinions utilizava uma escala de 1 a 5 estrelas nas avaliações.

A base de dados utilizada foi coletada em maio de 2011 ((TANG; GAO; LIU, 2012), (TANG *et al.*, 2012)) e contém, entre outros, a identificação do usuário e do item, avaliação, a data em que esta foi realizada, e também a lista de usuários de confiança para cada usuário listado.

A base original contém em torno de 1 milhão de avaliações de 300 mil produtos, feitos por 22 mil usuários. Neste trabalho, porém, utilizamos uma subamostra desta base, devido ao elevado número de modelos testados. Assim, trabalhamos com um total de 152.738 avaliações, 9.763 usuários e 12.009 produtos, com 55.180 relações de confiança direcionadas. A análise descritiva desta base pode ser encontrada no apêndice C.1.

A base foi particionada em base de Treinamento, com 90.000 avaliações, base de Validação, com 31.369 avaliações e base de Teste, também com 31.369 avaliações.

6.4.2 Filmtrust

Filmtrust combina rede social online e avaliações de filmes (GOLBECK; HENDLER, 2006). As avaliações são feitas na escala de 0,5 a 4,0 (de 0,5 em 0,.5).

A base Filmtrust, contendo todas as avaliações do site, foi coletada em junho de 2011 (GUO; ZHANG; YORKE-SMITH, 2013) e consiste em 35.497 avaliações, 1.508 usuários e 2.071 filmes, com 1.632 relações de confiança direcionadas. A análise descritiva desta base pode ser encontrada no apêndice C.2.

Neste trabalho, particionamos esta base em três: base de Treinamento, com 21.200 avaliações, base de Validação, com 7.149 avaliações e base de Teste, com 7.148 avaliações.

6.4.3 Yelp

Yelp permite buscar serviços locais com base em avaliações feitas pelos usuários. Criado em 2004, atualmente tem 150 milhões de avaliações, e recebe mais de 70 milhões acessos por

mês. Além disso, o site possui uma rede social extremamente ativa. As avaliações estão na escala de 1 a 5 estrelas.

A base Yelp, com 5,2 milhões de avaliações de 175 mil estabelecimentos, feitos por 1,3 milhões de usuários, foi distribuída pela própria empresa, como parte da Rodada 11 do Desafio Yelp (YELP, 2018).

Tal como no caso da base Epinions, sorteamos uma subamostra da base Yelp. Assim, trabalhamos com um total de 268.508 avaliações, 14.130 usuários e 6.844 estabelecimentos, com 99.432 relações de confiança direcionadas. A análise descritiva desta base pode ser encontrada no apêndice C.1.

A base foi particionada em base de Treinamento, com 160.000 avaliações, e bases de Validação e de Teste, com 54.254 avaliações cada.

A análise descritiva da subamostra estudada pode ser encontrada no apêndice C.3.

6.4.4 DeviantArt

Além de usar bases de dados bastante conhecidas em estudos de sistemas de recomendação, também coletamos uma base de dados própria como parte deste estudo.

DeviantArt é uma comunidade online focada em arte e fotografia. Lançado em 2000, o site atualmente contém mais de 300 milhões de trabalhos, produzidos por uma comunidade de 40 milhões de artistas. A plataforma é aberta e permite que tanto artistas iniciantes e amadores, quanto profissionais, exibam, promovam e compartilhem o trabalho deles com a comunidade, e também recebam feedback, seja na forma de likes e/ou de comentários, de outros usuários.

DeviantArt permite que o usuário siga um outro usuário (“watch”). Esta relação é direcionada, isto é, não precisa ser retribuída pelo outro usuário. Além disso, o usuário pode curtir (“favourite”) um determinado trabalho de um outro artista. Os usuários também podem agrupar seus favoritos em coleções temáticas.

A base de dados de usuários do DeviantArt foi coletada no período entre abril e julho de 2018. Iniciamos a coleta de dados com um grupo de usuários ativos em uma das sub-galerias do site. No próximo passo, incluímos aqueles usuários cujos trabalhos tinham sido adicionados aos favoritos por vários dos artistas do grupo inicial. Repetimos este procedimento recursivamente. Para cada usuário, coletamos a lista de usuários que o seguem, a lista de usuários listados

por ele como “amigos” (esta lista compreende um sub-conjunto dos usuários seguidos). Além disso, coletamos a listagem completa dos trabalhos favoritos deste usuário, incluindo o título do trabalho, o autor dele, e a coleção na qual o usuário alocou cada trabalho.

Para não fugir do escopo deste estudo, convertemos os dados sobre trabalhos favoritos em avaliação de um usuário por outro, com base na fração de favoritos em relação ao total de trabalhos deste usuário listados na base. Os detalhes desta conversão podem ser encontrados no apêndice C.4.

A base de dados resultante contém 196.057 avaliações de 14.853 artistas por 769 usuários, com 37.349 relações sociais direcionadas entre estes usuários. A análise descritiva desta base pode ser encontrada no apêndice C.4. Esta base foi particionada em base de Treinamento, com 120.000 avaliações, base de Validação, com 38.028 avaliações e base de Teste, também com 38.029 avaliações.

6.4.5 Sumário das estatísticas relevantes

Na Tabela 1, estão resumidas algumas estatísticas relevantes das quatro bases de dados estudadas, tais como número de avaliações, itens e produtos.

Tabela 1 – Estatísticas relevantes

	Epinions	Filmtrust	Yelp	DeviantArt
Avaliações	152.738	35.497	268.508	196.057
Usuários	9.763	1.508	14.130	769
Itens	12.009	2.071	6.844	14.853
Avaliações / usuário	15,6	23,5	19,0	255,0
Avaliações / item	12,7	17,1	39,2	13,2
Usuários com amigos (%)	56,8	34,6	98,9	98,6
Amigos / usuário	5,7	1,1	7,0	48,6
Amigos com itens avaliados em comum (%)	4,9	88,1	44,0	95,5
Amigos dos amigos com itens avaliados em comum (%)	45,7	73,1	76,5	98,8

6.5 Resumo dos Modelos

Segue, na Tabela 2, um resumo dos modelos de fatoração de matrizes testados, com breve caracterização de cada modelo, referências para a seção na qual eles foram descritos, e

também para a respectiva regularização.

Tabela 2 – Resumo dos Modelos de Fatoração de Matrizes

Modelo	Regularização Social	Norma	Seção	Equação
Modelo B	Sem	F	3.3	3.4
Modelo C	Sem	L^1	3.4	3.7
Modelo D	Sem	L^1	3.4	3.8
Modelo 1	Fator do usuário: distância ao centro	F	4.1.1	4.1 e 4.2
Modelo 2	Fator do usuário: propagação de gostos	F	4.1.2	4.3, 4.4 e 4.14
Modelo 2*	Modelo 2, similaridade com amigos dos amigos	F	4.1.2 e 4.1.5	4.4
Modelo 3	Fator do usuário: propagação de gostos	L^1	4.1.3	4.5 e 4.6
Modelo 4	Viés do usuário: propagação de gostos	F	4.2	4.16

Já os três modelos de Redes Neurais estão listados abaixo, na Tabela 3, também com breve caracterização e referência para a seção na qual estão descritos mais detalhadamente:

Tabela 3 – Resumo dos Modelos de Redes Neurais

Modelo	Descrição	Seção
MLP	Modelo básico de Perceptron de Múltiplas Camadas	5.1
Modelo 5	Média das avaliações dos amigos	5.2.1
Modelo 6	Embedding da mediana das avaliações dos amigos	5.2.2
Modelo 7	Vetor de 0,1 identificando amigos do usuário	5.2.3

Os modelos destacados em negrito são aqueles que foram propostos neste trabalho, enquanto os demais são abordagens propostas na literatura.

6.6 Resultados

Nos experimentos abaixo, os valores de d e λ foram escolhidos por validação cruzada para o Modelo B. Usamos sempre $\alpha = \lambda$.

6.6.1 Base Epinions

No ajuste, consideramos $d = 5$. Foram utilizadas as seguintes constantes:

$$\gamma = 0,001\,,$$
$$\lambda = \alpha = 0,2\,. \tag{6.3}$$

Os resultados dos ajustes estão apresentados na Tabela 4. O desvio padrão para os EQMs obtidos variou entre 0,010 e 0,012.

O modelo de melhor desempenho para esta base foi o Modelo 3 sem similaridade (em negrito), atingindo o EQM de 1,192, uma queda de 0,247 (20 desvios padrões) em relação ao Modelo B. De modo geral, os modelos com uso da norma L^1 na regularização (Modelo C, Modelo D e Modelo 3) mostraram ajuste significativamente melhor do que os modelos que usam a norma de Frobenius.

Tabela 4 – Base Epinions: EQM_{teste} para modelos de Fatoração de Matrizes com Regularização Social

Modelo	Similaridade						
	Sem Sim	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6
Modelo B	1,438						
Modelo C	1,309						
Modelo D	1,297						
Modelo 1	1,342	1,392	1,399	1,393	1,392	1,392	
Modelo 2	1,306	1,383	1,399	1,398	1,386	1,388	1,268
Modelo 2*		1,317	1,339		1,362	1,387	
Modelo 3	**1,192**	1,276	1,296	1,293	1,278	1,281	
Modelo 4	1,368						

A similaridade parece ter pouco ou nenhum impacto positivo no resultado. O uso da similaridade pode, inclusive, levar a piora do resultado: isso pode ser observado no Modelo 2, para todas as similaridades com exceção da Sim_6. Esta foi a similaridade de melhor performance, beneficiando-se da introdução de mais um parâmetro. Isso se deve ao fato de que a maioria dos usuários da base não avaliou os mesmos produtos que os seus amigos, levando a um número elevado de similaridades iguais a 0. Assim, estendendo a similaridade para os amigos dos amigos, no Modelo 2, observamos um resultado ligeiramente melhor (por exemplo, para a Sim_1, o EQM caiu de 1,383 a 1,317, 5 desvios padrões), porém ainda inferior ao resultado obtido para o Modelo 2 sem similaridade (EQM de 1,306).

Considerando somente os modelos com similaridades de 1 a 5, a Sim_1 teve o melhor desempenho em todos os modelos, seguida de Sim_4, segunda melhor para 3 dos 4 modelos com similaridade. A Sim_2 teve o pior desempenho para 3 dos 4 modelos, com a notável exceção do modelo em que calculamos a similaridade com base nos itens avaliados pelo usuário, e pelos amigos dos amigos. Isso se deve ao fato de que esta similaridade é nula também quando os usuários avaliaram somente um item em comum. Quando trabalhamos com amigos de amigos, temos um número bem maior de itens avaliados, e portanto um número maior de itens avaliados em comum.

A inclusão dos amigos dos amigos para o cálculo da similaridade leva a diminuição do EQM para as quatro similaridades testadas. Para três delas, esta diminuição é significativa. O melhor resultado, observado para a Sim_1, de 1,317, é bastante próximo do EQM obtido para o Modelo 2 sem similaridade, de 1,306. Isso se deve ao fato de que, enquanto a maioria absoluta dos pares de amigos não avaliou itens em comum, entre os amigos de amigos a fração de pares com avaliações em comum sobe para 46%.

A regularização social para o viés do usuário (Modelo 4) levou, neste caso, a uma diminuição significativa do EQM, para 1,368, uma queda de 5 desvios padrões em relação ao modelo B.

Agora, vamos dar uma olhada nos modelos de Filtragem Colaborativa Neural, mostrados na Tabela 5.

Tabela 5 – Base Epinions: EQM_{teste} para modelos de Filtragem Colaborativa Neural

Modelo	MLP	Modelo 5	Modelo 6	Modelo 7
EQM_{teste}	1,165	1,085	1,182	1,105

Todos os modelos de Filtragem colaborativa Neural têm desempenho superior ao dos modelos tradicionais, tanto em relação ao Modelo B, quanto em relação ao modelo de Filtragem Colaborativa de melhor desempenho. Os Modelos 5 e 7 produziram resultados significativamente melhores do que o do MLP simples, confirmando o potencial da inclusão das informações da rede social no sistema de recomendação.

6.6.2 Base Filmtrust

No ajuste, consideramos $d = 15$. Foram utilizadas as seguintes constantes:

$$\gamma = 0,001\,,$$
$$\lambda = \alpha = 0,3\,. \tag{6.4}$$

Os resultados dos ajustes estão apresentados na Tabela 6. Nesta tabela, optamos por omitir os ajustes do Modelo 3, já que estes se mostraram muito inferiores aos demais, produzindo ajustes com EQM similar aos do Modelo C e Modelo D.

Tabela 6 – Base Filmtrust: EQM_{teste} para modelos de Fatoração de Matrizes com Regularização Social

Modelo	Similaridade						
	Sem Sim	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6
Modelo B	0,822						
Modelo C	1,128						
Modelo D	1,174						
Modelo 1	0,797	0,802	0,806	0,802	0,801	0,802	
Modelo 2	0,798	0,795	0,802	0,805	0,797	0,799	0,807
Modelo 2*		0,799	0,802		0,802	0,802	
Modelo 4	0,816						

O desvio padrão para os EQMs variou entre 0,011 e 0,013.

Neste caso, ao contrário da base Epinions, a norma L^1 teve um desempenho péssimo, significativamente inferior ao dos modelos que utilizam a norma de Frobenius.

Apesar de que praticamente todos os modelos com regularização social mostraram uma melhora significativa em relação ao Modelo B, não temos um modelo que se destaca claramente como melhor.

Observamos que a diferença de performance entre os modelos com e sem similaridade é menor neste exemplo, o que se deve ao fato de que quase 90% dos pares de amigos têm filmes avaliados em comum, de modo que podemos avaliar a similaridade para a maioria dos pares.

O modelo com regularização social para o viés do usuário não levou a uma melhora significativa do ajuste.

Na Tabela 7, estão apresentados os valores de EQM_{teste} dos modelos de Filtragem Colaborativa Neural. Neste caso, ao contrário do que observamos para a base Epinions, estes

Tabela 7 – Base Filmtrust: EQM_{teste} para modelos de Filtragem Colaborativa Neural

Modelo	MLP	Modelo 5	Modelo 6	Modelo 7
EQM_{teste}	0,813	0,815	0,823	0,812

modelos não levam a uma melhora significativa não só em relação ao modelo de menor EQM, como também em relação ao Modelo B.

6.6.3 Base Yelp

No ajuste, consideramos $d = 5$. Foram utilizadas as seguintes constantes:

$$\gamma = 0,001\,,$$
$$\lambda = \alpha = 0,2\,. \tag{6.5}$$

Tabela 8 – Base Yelp: EQM_{teste} para modelos de Fatoração de Matrizes com Regularização Social

Modelo	Similaridade						
	Sem Sim	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6
Modelo B	1,089						
Modelo C	0,997						
Modelo D	0,968						
Modelo 1	1,038	1,046	1,076	1,046	1,046	1,047	
Modelo 2	**0,944**	1,023	1,075	1,071	1,032	1,038	**0,941**
Modelo 2*		**0,935**	**0,934**		**0,938**	**0,936**	
Modelo 3	**0,938**	**0,945**	0,962	0,959	**0,946**	**0,948**	
Modelo 4	1,009						

O desvio padrão para os EQMs variou entre 0,006 e 0,007. Os resultados estão apresentados na Tabela 8.

O Modelo 1 teve um desempenho muito inferior ao dos modelos 2 e 3. O modelo de melhor desempenho foi o Modelo 2*, com EQM de 0,934 para a Sim_2 (22 desvios padrões em relação ao Modelo B). Na tabela acima, este modelo - e também outros que apresentaram resultados comparáveis (a diferença entre estes resultados não é significativa), estão destacados em negrito.

Assim, podemos destacar os modelos 2 e 3 sem similaridade, e também o bom desempenho da Sim_1 e Sim_6. A Sim_2, tal como para a base Epinions, teve excelente desempenho quando

calculada para os amigos dos amigos, e o pior desempenho entre todas as similaridades para os demais modelos.

O uso do modelo com regularização social para o viés do usuário (Modelo 4) levou, para a base Yelp, a uma diminuição significativa do EQM, de 1,089 para 1,009, uma queda de 11 desvios padrões em relação ao modelo B.

Tabela 9 – Base Yelp: EQM_{teste} para modelos de Filtragem Colaborativa Neural

Modelo	MLP	Modelo 5	Modelo 6	Modelo 7
EQM_{teste}	0,925	0,898	0,947	0,905

O modelo de Perceptron de Múltiplas Camadas (Tabela 9)mostra um excelente desempenho para a base Yelp, com EQM similar ao do melhor modelo de fatoração de matrizes com Regularização Social (a diferença de 0,009 entre os dois modelos não é significativa).

Tal como para a base Epinions, os Modelos 5 e 6 levam a um incremento significativo de EQM em relação ao modelo MLP. Mesmo a inclusão simples e imperfeita das informações da rede social leva a uma melhora na qualidade do ajuste.

6.6.4 Base DeviantArt

No ajuste, consideramos $d = 5$. Foram utilizadas as seguintes constantes:

$$\gamma = 0,001\,,$$
$$\lambda = \alpha = 0,2\,. \tag{6.6}$$

Tabela 10 – Base DeviantArt: EQM_{teste} para modelos de Fatoração de Matrizes com Regularização Social

Modelo	Similaridade						
	Sem Sim	Sim_1	Sim_2	Sim_3	Sim_4	Sim_5	Sim_6
Modelo B	0,386						
Modelo C	0,391						
Modelo D	0,384						
Modelo 1	0,380	0,380	0,379	0,379	0,380	0,380	
Modelo 2	0,375	0,374	0,374	0,378	0,374	0,374	0,376
Modelo 2*		0,375	0,376		0,378	0,379	
Modelo 3	0,375	0,375	0,375	0,374	0,375	0,374	
Modelo 4	0,398						

O desvio padrão para os EQMs variou entre 0,003 e 0,004. Os resultados podem ser vistos na Tabela 10.

Para a base do DeviantArt, a melhora de performance devido à introdução da regularização social foi pequena. Isso se deve, provavelmente, ao fato de que temos um número muito grande de avaliações por usuário, o que permite fazer estimativas boas mesmo sem apelar às informações dos amigos de um certo usuário. Uma segunda explicação possível é que a inferência das avaliações a partir de avaliações implícitas pode ter resultado em uma maior variabilidade.

O uso dos Modelos 2, 2* e 3 leva a uma queda significativa no EQM, porém não há diferença significativa entre estes modelos, e também entre suas versões com diferentes similaridades.

Finalmente, tal como para a base FilmTrust, o modelo com regularização social para o viés do usuário não levou a uma melhora significativa do ajuste.

Tabela 11 – Base DeviantArt: EQM_{teste} para modelos de Filtragem Colaborativa Neural

Modelo	MLP	Modelo 5	Modelo 6	Modelo 7
EQM_{teste}	0,381	0,368	0,382	0,357

Na Tabela 11 podemos ver que o modelo MLP não traz uma diminuição significativa do EQM em comparação com o Modelo B. Contudo os Modelos 5 e 7 têm um excelente desempenho, com melhora significativa quanto em relação ao modelo B, tanto em relação ao melhor modelo com Regularização Social.

6.6.5 Resumo

Na Figura 9, está resumido o desempenho de alguns modelos que, de alguma forma, se destacaram neste estudo. Cada barra corresponde ao intervalo de confiança de 95%. Em azul estão os modelos sem regularização social, com os modelos propostos neste trabalho destacados em azul mais claro. Em vermelho e laranja, estão os modelos com regularização social, sendo em laranja os modelos que foram propostos neste trabalho.

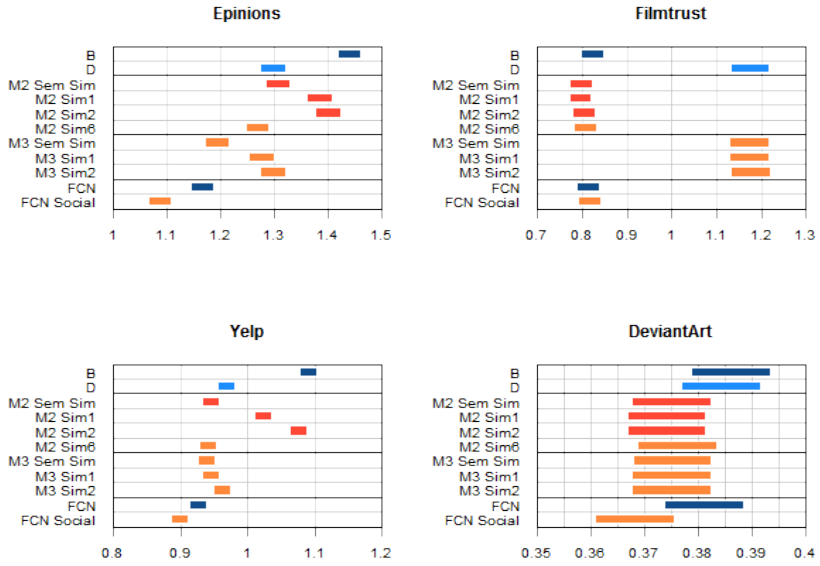

Figura 9 – Desempenho de alguns modelos em destaque

Entre os modelos presentes na figura estão os Modelos B e D, e também Modelos 2 e 3. O modelo de Filtragem Colaborativa Neural Social apresentado é o Modelo 5.

CAPÍTULO

7

DISCUSSÃO E CONCLUSÕES

Com base na intuição de que as informações sobre as preferências dos amigos do usuário em uma rede social podem melhorar a capacidade preditiva dos Sistemas de Recomendação, estudamos os modelos propostos em Ma *et al.* (2011), e também propusemos três modelos novos para a incorporação da regularização social aos Sistemas de Recomendação por fatoração de matrizes. Além disso, foram propostas e testadas quatro novas funções de similaridade, além das duas que foram estudadas em Ma *et al.* (2011). Finalmente, também implementamos o Modelo B, que é uma solução bem estabelecida para este tipo de problema, e largamente usado em produção, para usar como referência.

Com base na análise experimental, com aplicação dos modelos propostos em quatro bases de dados de tamanho razoável, podemos concluir que o uso de informações da rede social melhora significativamente o poder de predição do Sistema de Recomendação. Esta melhora pode ser pequena, como no caso das bases DeviantArt e Filmtrust (veja as seções 6.6.4 e 6.6.2 respectivamente), ou chegar a uma queda de 10-15% do EQM em relação ao modelo sem regularização social (seções 6.6.1 e 6.6.3), mas em todos os casos ela foi significativa.

Entre os modelos testados, os de melhor desemprenho, de modo geral, foram os Modelos 2 e 3.

Já entre as similaridades, as que se destacaram foram a Sim_1, Sim_2 e Sim_6. Em maioria dos casos, modelos com similaridade tiveram performance inferior, especialmente em bases de dados com menor número de amigos e de itens avaliados por usuário. Trata-se de um problema de

esparsidade dos dados similar ao enfrentado em métodos de vizinhança de Filtragem Colaborativa: temos um número muito grande de itens, e cada usuário avalia uma fração pequena destas, de modo que a probabilidade da intersecção entre o conjunto dos itens avaliados pelo usuário A e o conjunto avaliado pelo usuário B não ser vazia é muito baixa. No caso de uma base de dados em que o usuário médio tem poucos amigos e poucos pares de amigos avaliaram os mesmos itens, usar um modelo sem similaridade é a melhor opção (o caso da base Epinions, por exemplo). Neste caso, também é possível usar a similaridade calculada sobre o grupo dos amigos dos amigos. Em particular, recomenda-se o uso da Sim_2 somente quando trabalhamos com amigos dos amigos, pois esta similaridade exige um número maior de itens avaliados em comum.

Vale destacar que o índice de Jaccard (Sim_3), apesar de não ter uma performance tão boa quanto as demais, tem a vantagem de não depender das avaliações feitas pelos usuários, mas somente do ato de avaliação de um itens. Assim, ela pode ser usada em uma variedade maior de situações, por exemplo quando uma avaliação consiste na compra ou na visualização do item pelo usuário.

Também em dois dos quatro exemplos (seções 6.6.1 e 6.6.3), o Modelo 4 levou a uma queda significativa de EQM, mostrando o potencial deste modelo em aplicações práticas, possivelmente usando conjuntamente a regularização social para o fator do usuário e também para o viés do usuário.

O modelo de MLP, embora não seja uma solução universal (para a base Filmtrust, este modelo teve desempenho pior do que os modelos de fatoração de matrizes, mesmo sem Regularização Social, e para a base DeviantArt, desempenho comparável ao do Modelo B), mostrou excelente performance em duas das quatro bases testadas, base Epinions e base Yelp. Considerando as vantagens deste modelo, tais como sua grande flexibilidade e a velocidade de execução, a sua performance confirma o elevado potencial dos modelos que fazem uso de Redes Neurais em Sistemas de Recomendação.

Os outros modelos de Filtragem Colaborativa Neural, especificamente modelos 5 e 7, mostraram, para três dos quatro exemplos estudados, um desempenho superior ao MLP puro. Mesmo se tratando de uma incorporação imperfeita, registramos melhora significativa. Isso confirma o potencial dos estudos para melhor integração das informações da rede social nos modelos de Filtragem Colaborativa Neural.

7.1 Conclusões

O Modelo D, proposto neste trabalho, teve uma performance significativamente melhor que o Modelo B, de referência, em duas das quatro bases de dados testadas (Yelp e Epinions), e igual ao Modelo B na terceira (DeviantArt). Analogamente, o Modelo 3 teve desempenho igual ou superior aos Modelos 1 e 2 em três das quatro bases de dados testadas.

As similaridades usadas na literatura foram as que tiveram melhor desempenho de modo geral. Entre as similaridades propostas neste trabalho, somente a Sim_6 superou-as.

De modo geral, incorporação das informações da Rede Social leva a uma precisão maior na estimação das avaliações, tanto em modelos de Fatoração de Matrizes, quanto na Filtragem Neural colaborativa.

Contudo, para que este ganho de fato ocorra, precisamos de uma Rede Social suficientemente desenvolvida. Em particular, em situações em que os usuários têm poucos amigos (base Filmtrust, por exemplo), a incorporação das informações da rede social traz pouco ou nenhum benefício. Uma segunda limitação é o número de itens avaliado em comum por pares de amigos. Quando trabalhamos com milhões de produtos, a tendência é que este número seja baixo, ou mesmo nulo, o que inviabiliza o uso de similaridades.

Finalmente, a principal limitação é a necessidade da existência de uma rede social interna. Devido a dificuldade de capturar e validar os dados, de modo a garantir que o usuário *c* da nossa base é exatamente o usuário *c* da rede social, é praticamente impossível trazer informações de uma Rede Social externa para agregar a um Sistema de Recomendação.

7.2 Trabalhos Futuros

De modo geral, como pudemos ver, o uso das informações das redes sociais permite melhorar - em alguns casos, melhorar muito - as estimativas das avaliações, indicando o potencial desta abordagem. Em particular, além do uso das relações sociais somente, futuros estudos poderiam focar no uso também do conteúdo da rede social. Por exemplo, em vez de estabelecer similaridades entre usuários com base em avaliações feitas, poderia-se explorar as possibilidades do uso de outras informações, tais como posts ou perfis, para identificar usuários similares.

Fica a explorar também a identificação de usuários influenciadores, ou de bots em redes sociais, e o impacto da sua presença na regularização social.

7.2.1 Fatoração de Matrizes

O bom desempenho da similaridade Sim_6 levanta a questão sobre as constantes de regularização. Parece promissora a abordagem de tratar estas constantes como parâmetros a serem estimados pelo próprio modelo via Gradiente Descendente, e não como constantes. Em particular, esta abordagem resolveria a questão da validação cruzada em múltiplas dimensões, simplificando a escolha dos parâmetros ótimos.

Entre modelos de Fatoração de Matrizes estudados neste trabalho, se destacam o Modelo B, modelo de referência, e o Modelo D. As regularizações destes dois modelos poderiam ser descritas em termos de modelos Ridge e Lasso respectivamente. Assim, poderíamos pensar em uma regularização Elastic Net, combinando os dois, para um futuro estudo, incluindo ou não Regularização Social.

7.2.2 Filtragem Colaborativa Neural

Conforme vimos, tanto o potencial dos Sistemas de Recomendação baseados em Redes Neurais, quanto da incorporação das informações de Rede Social nestes sistemas é elevado. Uma possibilidade de exploração e estudo, nesse sentido, é se afastar da arquitetura de MLP. Assim, poderia ser interessante explorar as Redes Neurais Recorrentes para processar as informações da rede social. Redes LSTM (do inglês Long short-term memory) revolucionaram o reconhecimento de fala, e são muito usadas em áreas como tradução automática e modelagem de linguagem, justamente casos em que se trabalha com sequências e tamanho variável.

Um possível problema com esta abordagem é que, pelo menos no caso básico, a lista dos amigos de um usuário por si só não tem uma ordem inerente, de modo que precisamos de informação adicional (por exemplo, a data em que cada amigo foi adicionado), ou de alguma outra lógica de ordenação.

Outra possibilidade, talvez mais promissora, também emprestada da área de processamento de linguagem, é o uso de Redes Neurais Recursivas, em que o mesmo conjunto de pesos é aplicado recursivamente sobre uma entrada estruturada.

Em suma, há um vasto campo de possibilidades a serem exploradas no que diz respeito a uso de informações da Rede Social na Filtragem Neural Colaborativa, nas quais não focamos aqui para não se afastar demais do foco principal do trabalho. Contudo, estas possibilidades certamente merecem uma exploração mais cuidadosa.

REFERÊNCIAS

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge and Data Engineering**, v. 17, p. 734–749, 2005. Citado nas páginas 19 e 20.

AGGARWAL, C. C. **Recommender Systems**. [S.l.]: Springer International Publishing, 2016. Citado nas páginas 20, 21 e 25.

BARRAGÁNS-MARTÍNEZ, A. B.; COSTA-MONTENEGRO, E.; BURGUILLO, J. C.; REY-LÓPEZ, M.; MIKIC-FONTE, F. A.; PELETEIRO-RAMALLO, A. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. **Inf. Sci.**, v. 180, p. 4290–4311, 2010. Citado na página 19.

BELL, R. M.; KOREN, Y.; VOLINSKY, C. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In: **Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining**. New York, NY, USA: ACM, 2007. Citado na página 17.

BELLOGÍN, A.; CASTELLS, P.; CANTADOR, I. Improving memory-based collaborative filtering by neighbour selection based on user preference overlap. In: **Proceedings of the 10th Conference on Open Research Areas in Information Retrieval**. Paris, France: LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2013. Citado na página 20.

CACHEDA, F.; CARNEIRO, V.; FERNÁNDEZ, D.; FORMOSO, V. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. **TWEB**, v. 5, p. 2:1–2:33, 2011. Citado na página 21.

CHANG, T.-M.; HSIAO, W.-F. Model-based collaborative filtering to handle data reliability and ordinal data scale. **2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)**, v. 3, p. 2065–2069, 2011. Citado na página 20.

ECKART, C.; YOUNG, G. The approximation of one matrix by another of lower rank. **Psychometrika**, v. 1(3), p. 30–37, 1936. Citado na página 26.

FUNK, S. **Netflix Update: Try This at Home**. 2006. Disponível em: <http://sifter.org/~simon/journal/20061211.html>. Citado nas páginas 27, 28 e 45.

G.CAMPANA, M.; DELMASTRO, F. Recommender systems for online and mobile social networks: A survey. **Online Social Networks and Media**, v. 3-4, p. 75–97, 2017. Citado nas páginas 20, 21 e 33.

GOLBECK, J.; HENDLER, J. Filmtrust: movie recommendations using trust in web-based social networks. **CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.**, v. 1, p. 282–286, 2006. Citado na página 46.

GUO, G.; ZHANG, J.; YORKE-SMITH, N. A novel bayesian similarity measure for recommender systems. In: **Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)**. [S.l.]: AAAI Press, 2013. p. 2619–2625. Citado na página 46.

HE, X.; LIAO, L.; ZHANG, H.; NIE, L.; HU, X.; CHUA, T.-S. Neural collaborative filtering. In: **Proceedings of the 26th International Conference on World Wide Web**. Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017. Citado nas páginas 18, 22, 37 e 38.

HERLOCKER, J. L.; KONSTAN, J. A.; RIEDL, J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. **Information Retrieval**, v. 5, p. 287–310, 2002. Citado na página 21.

ILLIG, J.; HOTHO, A.; JÄSCHKE, R.; STUMME, G. A comparison of content-based tag recommendations in folksonomy systems. In: **KONT/KPP**. [S.l.: s.n.], 2007. Citado na página 19.

JACCARD, P. The distribution of the flora in the alpine zone. **New Phytologist**, v. 11, n. 2, p. 37–50, fev. 1912. Citado na página 34.

JANNACH, D.; ZANKER, M.; FELFERNIG, A.; FRIEDRICH, G. **Recommender systems: an introduction**. [S.l.]: Cambridge University Press, 2010. Citado na página 17.

KOREN, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. **Computer**, v. 42, p. 30–37, 2009. Citado na página 17.

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. **Computer**, v. 42(8), p. 30–37, 2009. Citado nas páginas 18 e 21.

LU, J.; WU, D.; MAO, M.; WANG, W.; ZHANG, G. Recommender system application developments: A survey. **Decision Support Systems**, v. 74, p. 12–32, 2015. Citado na página 17.

MA, H.; ZHOU, D.; LIU, C.; LYU, M. R.; KING, I. Recommender systems with social regularization. In: **Proceedings of the fourth ACM international conference on Web search and data mining**. New York, NY, USA: ACM, 2011. p. 287–296. Citado nas páginas 18, 23, 24, 30, 31, 32, 33, 34 e 57.

MASSA, P.; AVESANI, P. Trust-aware collaborative filtering for recommender systems. In: **Proceedings of the 2007 ACM conference on Recommender systems**. New York, NY, USA: ACM, 2007. p. 17–24. Citado nas páginas 18 e 23.

____. Trust metrics in recommender systems. In: **Computing with Social Trust**. [S.l.]: Springer International Publishing, 2009. Citado na página 23.

PHUKSENG, T.; SODSEE, S. Calculating trust by considering user similarity and social trust for recommendation systems. **2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)**, p. 1–6, 2017. Citado na página 23.

PRECHELT, L. Automatic early stopping using cross validation: quantifying the criteria. **Neural networks : the official journal of the International Neural Network Society**, v. 11 4, p. 761–767, 1998. Citado na página 44.

SALAKHUTDINOV, R.; MNIH, A. Probabilistic matrix factorization. In: **NIPS'07 Proceedings of the 20th International Conference on Neural Information Processing Systems**. [S.l.]: Curran Associates Inc., 2007. p. 1257–1264. Citado na página 17.

SALAKHUTDINOV, R.; MNIH, A.; HINTON, G. E. Restricted boltzmann machines for collaborative filtering. In: **ICML**. [S.l.: s.n.], 2007. Citado na página 22.

SHAMBOUR, Q.; LU, J. A trust-semantic fusion-based recommendation approach for e-business applications. **Decision Support Systems**, v. 54, p. 768–780, 2012. Citado na página 20.

SRIVASTAVA, N.; SALAKHUTDINOV, R. Multimodal learning with deep boltzmann machines. **Journal of Machine Learning Research**, v. 15, p. 2949–2980, 2012. Citado na página 38.

SU, X.; KHOSHGOFTAAR, T. M. A survey of collaborative filtering techniques. **Advances in artificial intelligence**, v. 2009:4, 2009. Citado na página 17.

SUN, Z.; HAN, L.; HUANG, W.; WANG, X.; ZENG, X.; WANG, M.; YAN, H. Recommender systems based on social networks. **J. Syst. Softw.**, Elsevier Science Inc., New York, NY, USA, v. 99, n. C, p. 109–119, jan. 2015. Citado na página 24.

TANG, J.; GAO, H.; HU, X.; LIU, H. Exploiting homophily effect for trust prediction. In: **WSDM '13 Proceedings of the sixth ACM international conference on Web search and data mining**. New York, NY, USA: ACM, 2013. p. 53–62. Citado nas páginas 18 e 23.

TANG, J.; GAO, H.; LIU, H. mTrust: Discerning multi-faceted trust in a connected world. In: ACM. **Proceedings of the fifth ACM international conference on Web search and data mining**. New York, NY, USA, 2012. p. 93–102. Citado na página 46.

TANG, J.; GAO, H.; LIU, H.; SARMA, A. D. eTrust: Understanding trust evolution in an online world. In: ACM. **Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining**. New York, NY, USA, 2012. p. 253–261. Citado na página 46.

TANG, J.; HU, X.; LIU, H. Social recommendation: a review. **Social Network Analysis and Mining**, v. 3, p. 1113–1133, 2013. Citado na página 22.

XIA, Z.; DONG, Y.; XING, G. Support vector machines for collaborative filtering. In: **ACM Southeast Regional Conference**. New York, NY, USA: ACM, 2006. Citado na página 21.

YELP. **Yelp Dataset, 2018**. 2018. <http://web.archive.org/web/20180620130255/https://www.yelp.com/dataset>. Acessado: 2018-06-20. Citado na página 47.

APÊNDICE

A

GRADIENTE DESCENDENTE: DERIVAÇÃO

Neste apêndice, estão apresentados os gradientes para os modelos estudados, de forma mais detalhada.

A.1 Modelos C e D

Para obter o gradiente destes modelos, usamos o erro $E_{c,p}$ dado pela Equação 3.3.

Para o Modelo C (Equação 3.7), determinamos o gradiente para as variáveis C_c, P_p, μ_c e μ_p. Nas equações a seguir, $sgn(x)$ é a função sinal:

$$\begin{aligned}
\frac{\partial L_C(C,P,\mu)}{\partial \mu_c} &= \sum_p I_{(c,p)\in K} sgn(R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p) + \lambda \cdot sgn(\mu_c)\,, \\
\frac{\partial L_C(C,P,\mu)}{\partial \mu_p} &= \sum_p I_{(c,p)\in K} sgn(R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p) + \lambda \cdot sgn(\mu_p)\,, \\
\frac{\partial L_C(C,P,\mu)}{\partial C_c} &= \sum_p I_{(c,p)\in K} sgn(R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p) P_p + \lambda \cdot sgn(C_c)\,, \\
\frac{\partial L_C(C,P,\mu)}{\partial P_p} &= \sum_p I_{(c,p)\in K} sgn(R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p) C_p + \lambda \cdot sgn(P_p)\,.
\end{aligned} \tag{A.1}$$

Assim, para cada par $(c,p) \in K$, atualizamos os parâmetros da seguinte forma:

$$\begin{aligned}
\mu_c^* &\leftarrow \mu_c - \gamma(sgn(E_{c,p}) + \lambda \cdot sgn(\mu_c))\,, \\
\mu_p^* &\leftarrow \mu_p - \gamma(sgn(E_{c,p}) + \lambda \cdot sgn(\mu_p))\,, \\
C_c^* &\leftarrow C_c - \gamma(sgn(E_{c,p}) \cdot P_p + \lambda \cdot sgn(C_c))\,, \\
P_p^* &\leftarrow P_p - \gamma(sgn(E_{c,p}) \cdot C_c + \lambda \cdot sgn(P_p))\,.
\end{aligned}$$

Analogamente, para o Modelo D (Equação 3.8), temos que:

$$\begin{aligned}
\frac{\partial L_D(C,P,\mu)}{\partial \mu_c} &= -\sum_p 2I_{(c,p)\in K}(R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p) - \lambda \cdot sgn(\mu_c)\,, \\
\frac{\partial L_D(C,P,\mu)}{\partial \mu_p} &= -\sum_p 2I_{(c,p)\in K}(R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p) - \lambda \cdot sgn(\mu_p)\,, \\
\frac{\partial L_D(C,P,\mu)}{\partial C_c} &= -\sum_p I_{(c,p)\in K}(R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p)P_p - \lambda \cdot sgn(C_c)\,, \\
\frac{\partial L_D(C,P,\mu)}{\partial P_p} &= -\sum_p I_{(c,p)\in K}(R_{c,p} - \mu_c - \mu_p - C_c \cdot P_p)C_p - \lambda \cdot sgn(P_p)\,.
\end{aligned} \tag{A.2}$$

Para cada par $(c,p) \in K$, atualizamos os parâmetros da seguinte forma:

$$\begin{aligned}
\mu_c^* &\leftarrow \mu_c + \gamma(E_{c,p} - \lambda \cdot sgn(\mu_c))\,, \\
\mu_p^* &\leftarrow \mu_p + \gamma(E_{c,p} - \lambda \cdot sgn(\mu_p))\,, \\
C_c^* &\leftarrow C_c + \gamma(E_{c,p} \cdot P_p - \lambda \cdot sgn(C_c))\,, \\
P_p^* &\leftarrow P_p + \gamma(E_{c,p} \cdot C_c - \lambda \cdot sgn(P_p))\,.
\end{aligned}$$

A.2 Modelo 1

Neste modelo (Equações 4.1 e 4.2), a parte de regularização social depende somente de C_c. Dessa forma, o gradiente em P_p, μ_c e μ_p dos dois modelos será igual ao gradiente do Modelo B. Consequentemente, os passos para a atualização de P_p, μ_c e μ_p serão os mesmos que no Modelo B:

$$\begin{aligned}
\mu_c^* &\leftarrow \mu_c + \gamma(E_{c,p} - \lambda\ \mu_c)\,, \\
\mu_p^* &\leftarrow \mu_p + \gamma(E_{c,p} - \lambda\ \mu_p)\,, \\
P_p^* &\leftarrow P_p + \gamma(E_{c,p} \cdot C_c - \lambda\ P_p)\,.
\end{aligned} \tag{A.3}$$

Resta desenvolver a parte específica correspondente a cada um dos modelos propostos: o passo para atualizar C_c. Vamos trabalhar primeiro com o modelo $L_{1S}^*(C,P,\mu)$ a equação 4.2. Precisamos determinar o último elemento do gradiente.

Derivamos em C_c:

$$\begin{aligned}\frac{\partial L^*_{1S}(C,P,\mu)}{\partial C_c} &= \frac{\partial L_B(C,P,\mu)}{\partial C_c} + 2\alpha\left(C_c - \frac{\sum\limits_{c_2\in\mathbb{F}(c)} Sim(c,c_2)C_{c_2}}{\sum\limits_{c_2\in\mathbb{F}(c)} Sim(c,c_2)}\right) \\ &\quad - 2\alpha\left(\sum_{j\in\mathbb{F}^-(c)} \frac{Sim(c,j)\left(C_j - \frac{\sum\limits_{c_2\in F(j)} Sim(j,c_2)C_{c_2}}{\sum\limits_{c_2\in\mathbb{F}(j)} Sim(j,c_2)}\right)}{\sum\limits_{c_2\in\mathbb{F}(j)} Sim(j,c_2)}\right) \\ &= -2\left(\sum_p I_{(c,p)\in K}E_{c,p}P_p - \lambda C_c\right) \\ &\quad + 2\alpha\left(C_c - \frac{\sum\limits_{c_2\in\mathbb{F}(c)} Sim(c,c_2)C_{c_2}}{\sum\limits_{c_2\in\mathbb{F}(c)} Sim(c,c_2)} - \sum_{j\in\mathbb{F}^-(c)} \frac{Sim(c,j)\left(C_j - \frac{\sum\limits_{c_2\in F(j)} Sim(j,c_2)C_{c_2}}{\sum\limits_{c_2\in\mathbb{F}(j)} Sim(j,c_2)}\right)}{\sum\limits_{c_2\in\mathbb{F}(j)} Sim(j,c_2)}\right)\end{aligned} \tag{A.4}$$

Observe que, para obter o gradiente do Modelo 1 sem similaridade, basta tomar $Sim(c_i,c_j) = 1$ para quaisquer c_i e c_j.

A.3 Modelo 2

Tal como no Modelo 1, os passos para a atualização de P_p, μ_c e μ_p do Modelo 2 (Equações 4.3 e 4.4 serão os mesmos que no Modelo B, pois a regularização social depende de C_c somente.

Vamos trabalhar com a equação 4.4 para encontrar a derivada em C_c:

$$\frac{\partial L^*_{2S}}{\partial C_c} = -2\left(\sum_p I_{(c,p)\in K}E_{c,p}C_c - \lambda P_p - \alpha\sum_{c_2\in\mathbb{F}(c)} Sim(c,c_2)(C_c - C_{c_2}) + \alpha\sum_{j\in\mathbb{F}^-(c)} Sim(c,j)(C_c - C_j)\right). \tag{A.5}$$

Novamente, observamos que a regularização $L^*_2(C,P,\mu)$ é igual à regularização $L^*_{2S}(C,P,\mu)$ quando $Sim(c_1,c_2) = 1$ para todos os pares (c_1,c_2). Assim, obtemos o gradiente para $L^*_2(C,P,\mu)$ a partir da equação acima:

$$\frac{\partial L^*_2}{\partial C_c} = -2\left(\sum_p I_{(c,p)\in K}E_{c,p}C_c - \lambda P_p - \alpha\sum_{c_2\in\mathbb{F}(c)} (C_c - C_{c_2}) + \alpha\sum_{j\in\mathbb{F}^-(c)} (C_c - C_j)\right). \tag{A.6}$$

A.4 Modelo 3

No método do gradiente descendente, os passos para a atualização de P_p, μ_c e μ_p para o Modelo 3 (Equações 4.5 e 4.6 serão os mesmos que no Modelo C ou no Modelo D (dependo de qual modelo será utilizado), pois a regularização social depende de C_c somente.

Vamos trabalhar com a equação 4.6 para encontrar a derivada em C_c:

$$\frac{\partial L^*_{3S}}{\partial C_c} = \frac{\partial L_{C/D}}{\partial C_c} + 2\alpha \left(\sum_{c_2 \in \mathbb{F}(c)} Sim(c,c_2) sgn(C_c - C_{c_2}) - \sum_{j \in \mathbb{F}^-(c)} Sim(c,j) sgn(C_c - C_j) \right). \tag{A.7}$$

A regularização $L^*_3(C,P,\mu)$ é igual à regularização $L^*_{3S}(C,P,\mu)$ quando $Sim(c_1,c_2) = 1$ para todos os pares $(c_1,c_2$. Assim, obtemos o gradiente para $L^*_3(C,P,\mu)$:

$$\frac{\partial L^*_3}{\partial C_c} = \frac{\partial L_{C/D}}{\partial C_c} + 2\alpha \left(\sum_{c_2 \in \mathbb{F}(c)} sgn(C_c - C_{c_2}) - \sum_{j \in \mathbb{F}^-(c)} sgn(C_c - C_j) \right). \tag{A.8}$$

A.5 Descida pelo gradiente: Modelos com Similaridade Exponencial

Para encontrar o gradiente do Modelo 2 com Similaridade Exponencial (Equação 4.14), derivamos em β:

$$\frac{\partial L^*_{2S\beta}}{\partial \beta} = -\alpha \sum_{c \in V_c} \sum_{c_2 \in \mathbb{F}(c)} e^{-\beta S(c,c_2)} S(c,c_2) ||C_c - C_{c_2}||^2_F + 2\alpha\beta \,. \tag{A.9}$$

O cálculo do gradiente para o Modelo 3 é muito similar ao do Modelo 2.

Já para o Modelo 1, temos que:

$$L^*_{1S\beta}(C,P,\mu) = L_B(C,P,\mu) + \alpha \sum_{c \in V_c} \left|\left| C_c - \frac{\sum_{c_2 \in \mathbb{F}(c)} C_{c_2} e^{-\beta S(c,c_2)}}{\sum_{c_2 \in \mathbb{F}(c)} e^{-\beta S(c,c_2)}} \right|\right|^2_F . \tag{A.10}$$

Derivando em β, obtemos:

$$\frac{\partial L^*_{1S\beta}}{\partial \beta} = 2\alpha \sum_{c \in V_c} G(c,c_2) \left|\left| C_c - \frac{\sum_{c_2 \in \mathbb{F}(c)} C_{c_2} e^{-\beta S(c,c_2)}}{\sum_{c_2 \in \mathbb{F}(c)} e^{-\beta S(c,c_2)}} \right|\right|_F , \tag{A.11}$$

onde

$$G(c,c_2) = \frac{\sum\limits_{c_2 \in \mathbb{F}(c)} S(c,c_2) C_{c_2} e^{-\beta S(c,c_2)} \sum\limits_{c_2 \in \mathbb{F}(c)} e^{-\beta S(c,c_2)} - \sum\limits_{c_2 \in \mathbb{F}(c)} C_{c_2} e^{-\beta S(c,c_2)} \sum\limits_{c_2 \in \mathbb{F}(c)} S(c,c_2) e^{-\beta S(c,c_2)}}{\left(\sum\limits_{c_2 \in \mathbb{F}(c)} e^{-\beta S(c,c_2)} \right)^2}. \tag{A.12}$$

A.6 Modelo 4

Para o Modelo 4 sem similaridade (Equações 4.16, o gradiente de L_v em C_c, P_p e μ_p é o mesmo de $L(C,P,\mu)$, pois a regularização depende somente de μ_c.

Já o gradiente em μ_c é igual a:

$$\frac{\partial L_V^*}{\partial \mu_c} = \frac{\partial L(C,P,\mu)}{\partial \mu_c} + 2\alpha \left(\sum_{c_2 \in \mathbb{F}(c)} |\mu_c - \mu_{c_2}| - \sum_{j \in \mathbb{F}^-(c)} |\mu_c - \mu_j| \right). \tag{A.13}$$

Para o Modelo 4 com similaridade, o gradiente em μ_c é dado por:

$$\frac{\partial L_{v1}^*}{\partial \mu_c} = \frac{\partial L(C,P,\mu)}{\partial \mu_c} + 2\alpha \left(\sum_{c_2 \in \mathbb{F}(c)} Sim(c,c_2) |\mu_c - \mu_{c_2}| - \sum_{j \in \mathbb{F}^-(c)} Sim(c,j) |\mu_c - \mu_j| \right). \tag{A.14}$$

APÊNDICE

B

GRADIENTE DESCENDENTE ESTOCÁSTICO: CÁLCULO EM BLOCOS

Tradicionalmente, no método do Gradiente Descendente, varremos a base de observação em observação, atualizando os parâmetros a cada passo. Neste trabalho, utilizamos uma versão alternativa do algoritmo, em que o cálculo é feito com blocos de observações simultaneamente, visando maior velocidade de execução.

B.1 Avaliação a Avaliação: Algoritmo Básico

Na forma mais básica de cálculo, atualizamos as variáveis C_c, P_p, μ_c e mu_p para uma avaliação $R_{c,p}$ por vez, varrendo toda a base de avaliações. Depois, repetimos o procedimento até que o critério de parada seja satisfeito.

Algoritmo 1 – AVALIAÇÃO A AVALIAÇÃO

Entrada: Conjunto de n avaliações $R_{c,p}$, em que a i-ésima entrada R_{c_i,P_i} é a avaliação do item p_i pelo usuário c_i

Saída: C, P, μ_c e μ_p

repita

  para $i = 0;\ i < n;\ i++$ **faça**

    $E_{c_i,p_i} = R_{c_i,p_i} - \hat{R}_{c_i,p_i}$ Atualizamos C, P, μ_c, μ_p em função de $E(c_i, p_i)$;

  fim

até *que o critério de parada seja satisfeito*;

B.2 Avaliação a Avaliação: Blocos Aleatorizados

Podemos também atualizar as variáveis C_c, P_p, μ_c e mu_p para blocos de avaliações $R_{c,p}$ de tamanho n_b. Dessa forma, varremos a base toda, bloco a bloco. Como, agora, damos um passo bem grande de uma vez, e atualizamos simultaneamente vários parâmetros que, no algoritmo tradicional, seriam atualizados sequencialmente, o algoritmo não irá convergir para o mínimo.

Para resolver este problema, no final de cada passo, a base é reordenada ao acaso. Dessa forma, as avaliações contidas em cada bloco, e a sequência em que estas são usadas irá variar de um passo para o outro, e vamos avançar na direção correta.

Algoritmo 2 – AVALIAÇÃO A AVALIAÇÃO: BLOCOS ALEATORIZADOS

Entrada: Conjunto de n avaliações $R_{c,p}$, em que a i-ésima entrada R_{c_i,P_i} é a avaliação do produto p_i pelo usuário c_i
Parâmetro : n_b tamanho do bloco
Saída: C, P, μ_c e μ_p
repita
  para $i = 0;\ i < n;\ i = i + n_b$ **faça**
    R_{c_i,p_i} avaliações de i a $i + n_b - 1$;
    c_i usuários que fizeram estas avaliações (com possíveis repetições) ;
    p_i produtos que foram avaliados (com possíveis repetições) ;
    $E_{c_i,p_i} = R_{c_i,p_i} - \hat{R}_{c_i,p_i}$;
    Atualizamos C, P, μ_c, μ_p em função de $E(c_i, p_i)$;
  fim
  Reordenamos a base de avaliações ;
até *que o critério de parada seja satisfeito*;

APÊNDICE

C

ANÁLISE DESCRITIVA DAS BASES DE DADOS UTILIZADAS NO ESTUDO

Aqui, listamos um número de estatísticas descritivas referentes as bases de dados utilizadas que podem ser úteis para a compreensão da estrutura das bases e avaliação dos resultados obtidos.

C.1 Epinions

Para gerar a subamostra da base completa, utilizamos o seguinte procedimento. Num primeiro passo, amostramos ao acaso alguns usuários da base completa. Depois, completamos a amostra com amigos destes usuários escolhidos ao acaso. Feito isso, amostramos itens ao acaso entre aqueles que foram avaliados pelos usuários selecionados. Este procedimento foi adotado já que amostra aleatória simples, seja de trincas usuário - produto - avaliação, seja de usuários e produtos, resultou sempre em bases muito esparsas.

A subamostra da base de dados Epinions utilizada neste estudo contém 152.738 avaliações de 12.009 itens por 9.763 usuários, com 55.180 relações sociais direcionadas entre estes usuários. As avaliações variam entre 1 e 5, com média 3,96 e mediana 4.

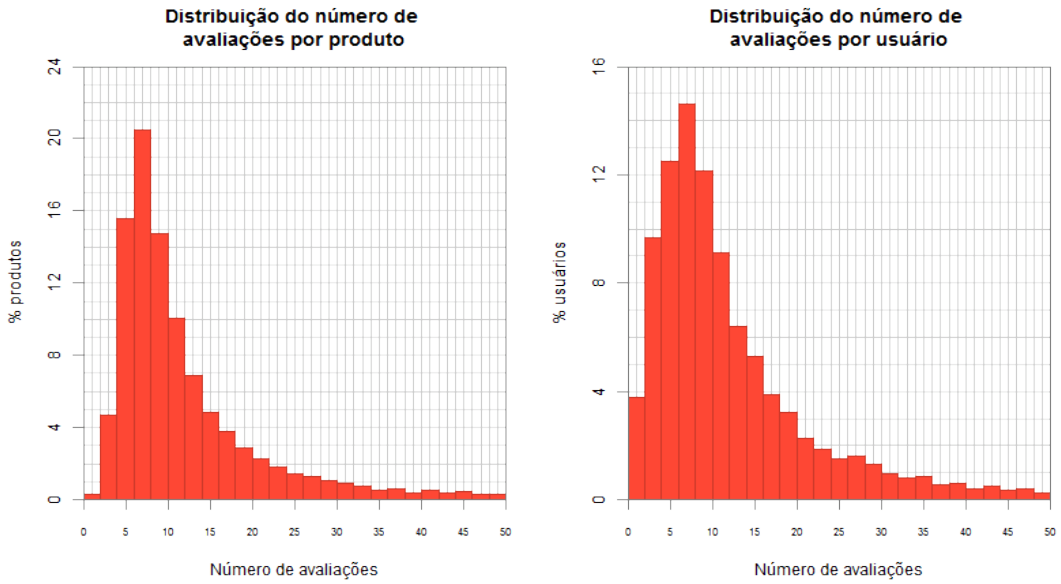

Figura 10 – Epinions: distribuição do número de avaliações por item e por usuário

Tabela 12 – Epinions: distribuição das avaliações

Avaliação	1	2	3	4	5
Contagem	10.156	12.391	18.474	44.473	67.244
%	6,7	8,1	12,1	29,1	44,0

Cada item foi avaliado, em média, 12,7 vezes e 2,7% dos itens receberam mais de 50 avaliações. O máximo de avaliações por item foi de 97. Já cada usuário avaliou, em média, 15,6 itens diferentes. O máximo de avaliações feitas por um usuário foi de 501, e 4,9% dos usuários avaliaram mais de 50 itens. A distribuição do número de avaliações por item e por usuário pode ser vista na Figura 10.

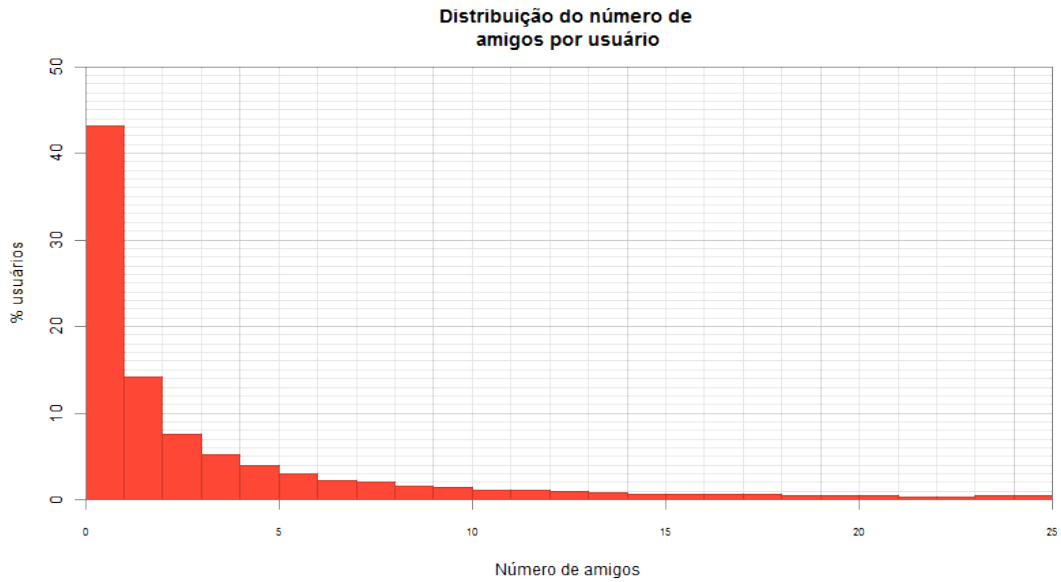

Figura 11 – Epinions: distribuição do número de amigos por usuário

Entre todos os usuários registrados na base, somente 56,9% têm pelo menos um amigo, e 5,7 % têm mais de 25 amigos. O maior número de amigos que um usuário tem é 196, enquanto o número médio de amigos por usuário é de 5,7.

Somente 4,9% dos pares de usuários ligados por relações de amizade têm ao menos um item avaliado em comum. Já se considerarmos o usuário e os amigos dos seus amigos, este percentual sobe para 45,7%.

C.2 Filmtrust

A base Filmtrust contém 35.497 avaliações de 2.071 itens por 1.508 usuários, com 1.632 relações sociais direcionadas entre os usuários. As avaliações variam entre 0,5 e 4,0, de 0,5 em 0,5, com média 3,00 e mediana 3,0.

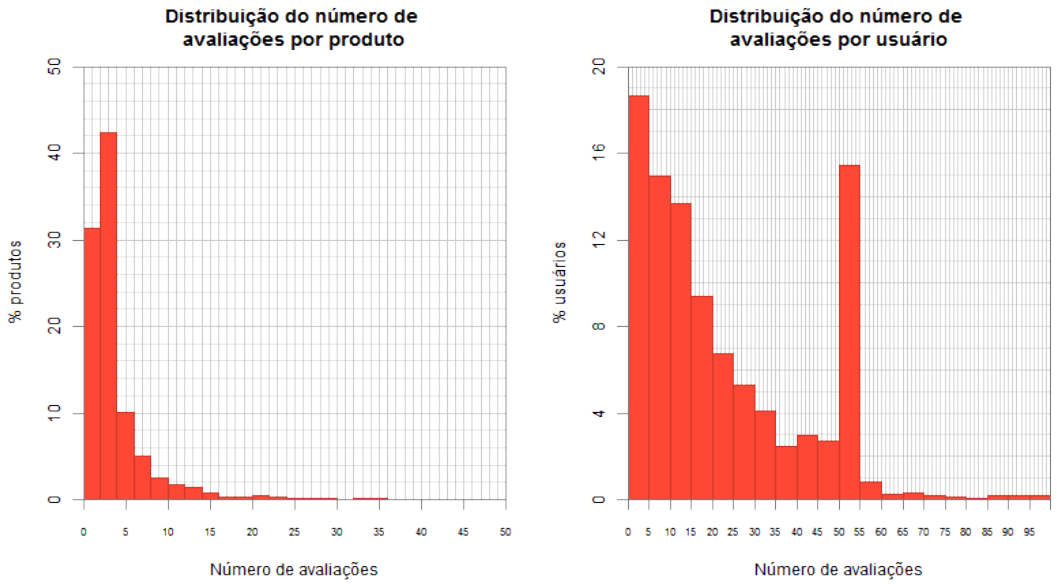

Figura 12 – Filmtrust: distribuição do número de avaliações por item e por usuário

Tabela 13 – Filmtrust: distribuição das avaliações

Avaliação	0,5	1,0	1,5	2,0	2,5	3,0	3,5	4,0
Contagem	1.060	1.141	1.601	3.113	4.392	7.877	7.142	9.171
%	3,0	3,2	4,5	8,8	12,4	22,2	20,1	25,8

Cada item foi avaliado, em média, 17,1 vezes e 2,6% dos itens receberam mais de 50 avaliações. O máximo de avaliações por item foi de 1044. Já cada usuário avaliou, em média, 23,5 itens diferentes. O máximo de avaliações feitas por um usuário foi de 244, e 1,1% dos usuários avaliaram mais de 100 itens. A distribuição do número de avaliações por item e por usuário pode ser vista na Figura 12.

Os usuários avaliaram, em particular, 50 melhores filmes de acordo com o American Film Institute. Estes foram avaliados pela maioria dos usuários. Além disso, muitos usuários avaliaram somente estes 50 filmes, o que explica o elevado percentual de usuários com exatamente 50 avaliações.

Entre todos os usuários, somente 34,6% têm pelo menos um amigo, e 0,3 % têm mais de 25 amigos. O maior número de amigos que um usuário tem é 57, enquanto o número médio de amigos por usuário é de 1,1.

Dos pares de usuários ligados por relações de amizade, 88,1% têm ao menos um item avaliado em comum. Este percentual elevado se deve, novamente, ao fato de que a maioria dos

usuários avaliou o mesmo conjunto de 50 filmes. Se considerarmos o usuário e os amigos dos seus amigos, este percentual cai para 73,1, devido ao baixo número de amigos por usuário.

C.3 Yelp

Para gerar a subamostra da base completa, utilizamos o seguinte procedimento. Num primeiro passo, amostramos ao acaso alguns usuários da base completa. Depois, completamos a amostra com amigos destes usuários escolhidos ao acaso. Feito isso, amostramos itens ao acaso entre aqueles que foram avaliados pelos usuários selecionados. Este procedimento foi adotado já que amostra aleatória simples, seja de trincas usuário - produto - avaliação, seja de usuários e produtos, resultou sempre em bases muito esparsas.

A subamostra da base de dados Yelp utilizada neste estudo contém 268.508 avaliações, 14.130 usuários e 6.844 itens, com 99.432 relações de confiança direcionadas. As avaliações variam entre 1 e 5, com média 3,87 e mediana 4.

Tabela 14 – Yelp: distribuição das avaliações

Avaliação	1	2	3	4	5
Contagem	11.579	20.935	48.487	97.093	90.414
%	4,3	7,8	18,0	36,2	33,7

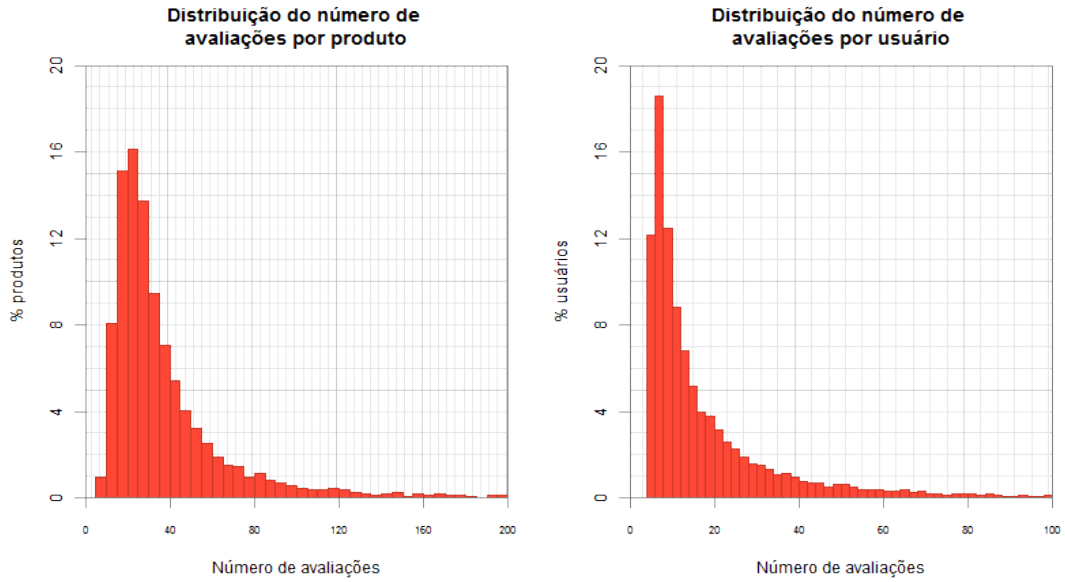

Figura 13 – Yelp: distribuição do número de avaliações por item e por usuário

Cada item foi avaliado, em média, 39,2 vezes e 0,9% dos itens receberam mais de 200 avaliações. O máximo de avaliações por item foi de 717. Cada usuário avaliou, em média, 19,0 itens diferentes. O máximo de avaliações feitas por um usuário foi de 327, e 1,4% dos usuários avaliaram mais de 100 itens. A distribuição do número de avaliações por item e por usuário pode ser vista na Figura 13.

Entre todos os usuários registrados na base, somente 1,2% não têm amigos registrados na base, e 2,6 % têm mais de 25 amigos. O maior número de amigos que um usuário tem é 59, enquanto o número médio de amigos por usuário é de 7,0.

Quase metade dos pares de usuários ligados por relações de amizade, mais precisamente 44,0%, têm ao menos um item avaliado em comum. Já se considerarmos o usuário e os amigos dos seus amigos, este percentual sobe para 76,5%.

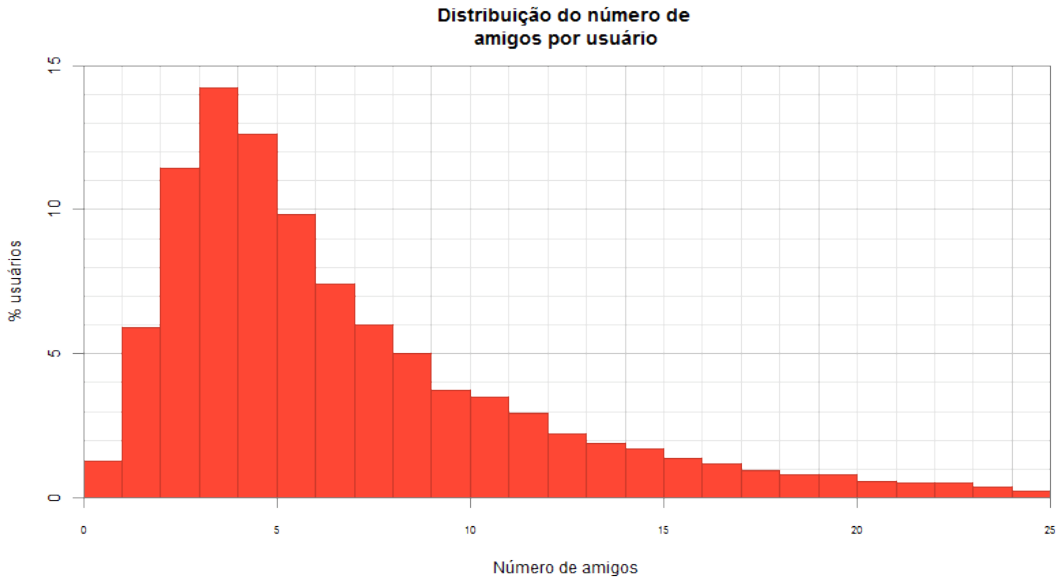

Figura 14 – Yelp: distribuição do número de amigos por usuário

C.4 DeviantArt

C.4.1 Conversão da base

A base de dados do DeviantArt, tal como foi coletada, contém informações sobre os favoritos (“favorites”, likes) somente. Já neste trabalho, trabalhamos com avaliações que assumem pelo menos dois valores. Assim, para não fugir do escopo, decidimos converter estas avaliações para uma escala diferente. Para isso, optamos por converter converter estas avaliações

em avaliações de usuários por outros usuários.

Primeiramente, desconsideramos os trabalhos dos usuários cujos favoritos foram coletados, obtendo, desta forma, dois grupos de usuários sem intersecção: aqueles que avaliaram os trabalhos (grupo dos avaliadores), e aqueles cujos trabalhos foram avaliados (grupo dos avaliados). Além disso, excluímos os usuários que deram menos de 100 ou mais de 5.000 mil favoritos (um deles com mais de 300.000), e também os usuários com menos de 20 ou mais de 1.000 obras. Os thresholds foram escolhidos com base na distribuição dos dados, de forma a remover os casos extremos, e também no comportamento típico dos usuários do site.

Feito isso, para cada par (c, p), onde c é o avaliador e p é o avaliado, calculamos f, a fração de trabalhos do artista p que foram favoritados pelo artista c. Convertemos f em avaliação $R_{c,p}$ usando a seguinte fórmula:

$$R_{c_p} = \begin{cases} 1 & \text{se } log(f) < -5.5 \\ 2 & \text{se } -5.5 \leq log(f) < -4.5 \\ 3 & \text{se } -4.5 \leq log(f) < -3.0 \\ 4 & \text{cc} \end{cases} \tag{C.1}$$

C.4.2 Estatísticas descritivas

A base DeviantArt contém 196.057 avaliações de 14.853 itens por 769 usuários, com 37.349 relações sociais direcionadas entre estes usuários. As avaliações variam entre 1 e 4, com média 2,47 e mediana 3.

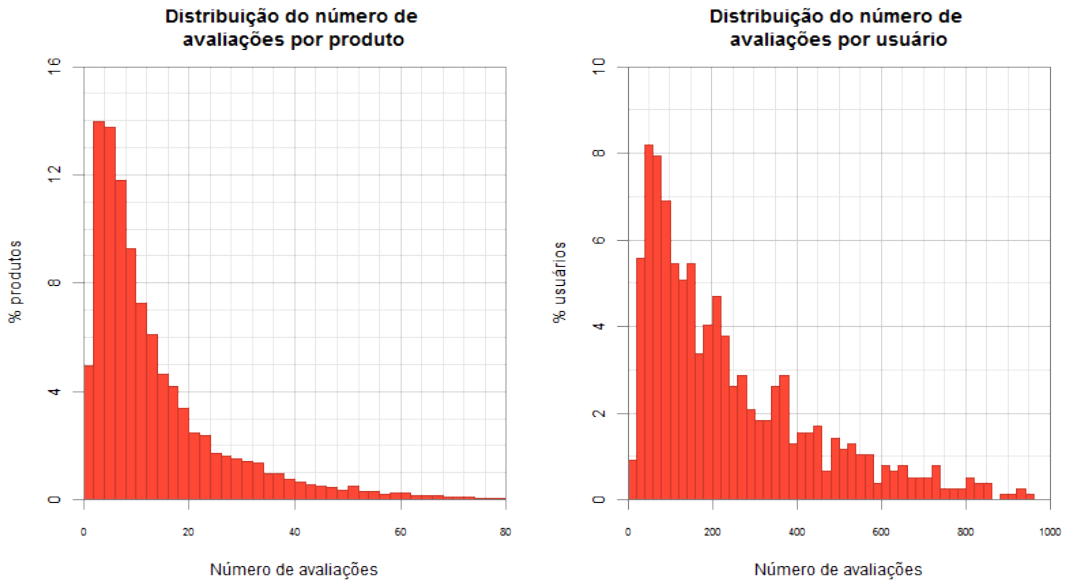

Figura 15 – DeviantArt: distribuição do número de avaliações por item e por usuário

Tabela 15 – DeviantArt: distribuição das avaliações

Avaliação	1	2	3	4
Contagem	25.540	69.224	84.732	16.561
%	13,0	35,3	43,2	8,5

Cada item foi avaliado, em média, 13,2 vezes e 0,3% dos itens receberam mais de 80 avaliações. O máximo de avaliações por item foi de 99. Já cada usuário avaliou, em média, 255,0 itens diferentes. O máximo de avaliações feitas por um usuário foi de 1253, e 1,6% dos usuários avaliaram mais de 100 itens. A distribuição do número de avaliações por item e por usuário pode ser vista na Figura 10.

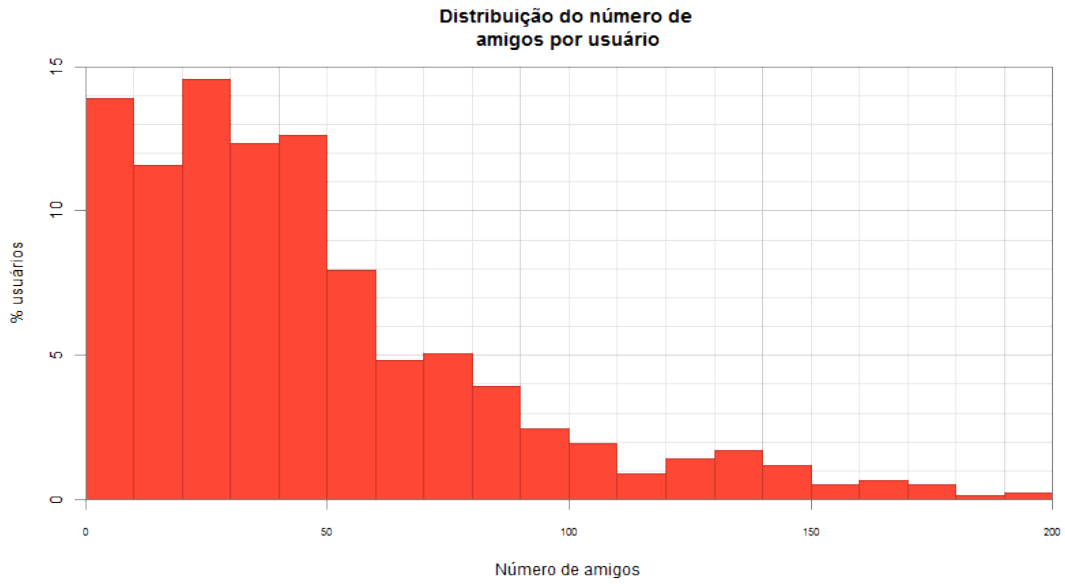

Figura 16 – DeviantArt: distribuição do número de amigos por usuário

Entre todos os usuários registrados na base, somente 98,6% têm pelo menos um amigo, e 1.6 % têm mais de 200 amigos. O maior número de amigos que um usuário tem é 285, enquanto o número médio de amigos por usuário é de 48,6.

Somente 4,5% dos pares de usuários ligados por relações de amizade não têm itens avaliados em comum. Já se considerarmos o usuário e os amigos dos seus amigos, este percentual cai para 1,2%.