

引用格式:熊孝全,张明新. 基于联合概率矩阵分解的微博关注推荐算法[J]. 微电子学与计算机, 2019, 36(8): 92 -96.
[XIONG X Q,ZHANG M X. Unified probabilistic matrix factorization of follow[J]. Microeletronics & Computer, 2019, 36(8): 92 -96.]

基于联合概率矩阵分解的微博关注推荐算法

熊孝全¹, 张明新²

(1 苏州大学 计算机科学与技术学院, 江苏 苏州 215006;
2 常熟理工学院 计算机科学与工程学院, 江苏 常熟 215500)

摘 要: 在传统的概率矩阵分解算法基础上,考虑交互行为的时效性建立用户影响力模型,并结合静态关注关系,提出了基于联合概率矩阵分解的微博关注推荐算法(UPMFF-L). 在使用 scrapy 爬取新浪微博得到的数据集上的实验结果表明,与 NMF、PMF 和 SoRec 算法相比,UPMFF-L 算法在不同数据密度下的 F1-Measure 平均提升了 11.82%.

关键词: 微博关注推荐;联合概率矩阵分解;关系相似度;影响力模型

中图分类号: TP301.6 **文献标识码:** A **文章编号:** 1000-7180(2019)08-0092-05

Unified probabilistic matrix factorization of follow

XIONG Xiao-quan¹, ZHANG Ming-xin²

(1 College of Computer Science and Technology, Soochow University, Suzhou 215006, China;
2 Department of Computer Science & Engineering, Changshu Institute of Technology, Changshu 215500, China)

Abstract: We add the timeliness of interaction behavior into the traditional probability matrix factorization algorithm to establish the newly proposed user influence model Combined with the static attention relationship, a recommendation algorithm based on unified probabilistic matrix factorization is proposed. The experimental results on the data set established by Sina Weibo show that compared with NMF, PMF, SoRec and other algorithms, the proposed algorithm has an average increase of 11.82% on the F1-Measure index under different data densities.

Key words: weibofollow recommendation; unified probabilistic matrix factorization; relationship similarity; influence model

1 引言

近年来,研究者将传统的推荐技术应用于社交网络推荐中.郭等人从关联关系出发,假设存在关联关系的对象更容易受到同一用户关注,利用对象间的关联关系进行推荐^[1]. Tang 等人认为在线社交网络中存在着不同于直接连接的弱依赖关系,同时指出社交网络具有异质性,并将两者结合起来应用于社交网络关注推荐中^[2]. 陈等人综合

考虑相似关系和信任关系筛选邻居节点进行推荐^[3]. 徐等人考虑了用户之间的交互行为,通过用户间的关注关系强度、交互频率、个体兴趣计算用户之间的关联兴趣,并基于此产生推荐列表^[4]. 上述方法仅考虑了用户之间的静态的关注关系和交互频率,并不能准确衡量用户之间的关系强度,导致推荐结果被接受概率低.

针对上述问题,本文在考虑交互行为的时效性特点的基础上,结合静态关注关系,提出了基于联合概率矩阵分解的微博关注推荐算法(Unified Proba-

bilistic Matrix Factorization of Follow, UPMFF).

2 微博关注推荐

微博关注推荐是将新的朋友推荐给用户,形成新的关注.为了提高准确率,需要推荐接受概率高的关注.通常,推荐被接受的概率与关系相似度以及推荐用户对其的影响力大小相关.对用户 u_i 、 u_j 和 u_k 来说, u_i 受 u_j 影响而关注 u_k 的概率用 p_{ik} 表示,定义如下:

$$p_{ik} = h(U_i^T V_j, U_i^T F_k) \quad (1)$$

式中, U_i 是用户 u_i 的隐含特征向量; V_j 是用户 v_j 的影响力隐含特征向量; F_k 是用户 u_k 的相似度隐含特征向量; $h(\cdot)$ 是参数 $U_i^T V_j$ 和 $U_i^T F_k$ 的函数.

为了避免噪声的影响,用户接受推荐的概率由用户、相似度和影响力潜在隐含特征向量得到.具体地,关系相似度由用户隐含特征向量和相似度隐含特征向量内积得到;影响力由用户隐含特征向量和影响力隐含特征向量内积得到.

在此基础上生成推荐列表,对 u_i 进行 Top-N 推荐,即选择列表中的前 N 个用户.

$$\text{Top}(u_i) = \arg \max_{b_u \in B}^N p_{ib} \quad (2)$$

3 基于联合概率矩阵分解的微博关注推荐算法

3.1 用户关系相似度

微博中的用户 u 和 v , 有如下关系:

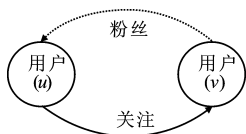


图 1 关注和粉丝关系

如果用户 u 关注了用户 v , 称 u 是 v 的关注者(follower), v 是 u 的粉丝(followee), u 对 v 是关注关系, v 对 u 是粉丝关系. 以这两种关系建立起的微博用户网络是一张有向图. 因此, 可以使用图中节点的相似度来表示关系相似度. 考虑节点共同邻居的数目, 节点相似度正比于共同邻居数目. 综合考虑节点出度相似度和入度相似度, 对两种相似度的计算公式如(3)(4)所示:

$$\text{Sim}_I(u, v) = \frac{|N_I(u) \cap N_I(v)|}{|N_I(u)| + |N_I(v)|} \quad (3)$$

$$\text{Sim}_O(u, v) = \frac{|N_O(u) \cap N_O(v)|}{|N_O(u)| + |N_O(v)|} \quad (4)$$

式中, $\text{Sim}_I(u, v)$ 是用户 u 和 v 的粉丝关系相似度; $\text{Sim}_O(u, v)$ 是 u 和 v 的的关注关系相似度; $N_I(u)$ 和 $N_I(v)$ 分别表示 u 和 v 的粉丝用户的集合; $N_O(u)$ 和 $N_O(v)$ 分别表示 u 和 v 的关注用户的集合.

文献[5]的实验结果表明双向关系对节点相似度的贡献并不相同. 因此, 使用双向关系线性加权来计算最终的关系相似度. 如下式所示:

$$\text{Sim}(u, v) = \lambda \text{Sim}_I(u, v) + (1 - \lambda) \text{Sim}_O(u, v) \quad (5)$$

式中, λ 是粉丝相似度的权值.

为了将相似度映射到 $[0, 1]$, 引入用户相似度权重, 如公式(6)所示:

$$w_i = \frac{\text{Sim}(u_i, v)}{\sum_{j=1}^n \text{Sim}(u_j, v)} \quad (6)$$

式中, w_i 是用户 u_i 与 v 的关系相似度在 v 所有相似用户中的权重; n 是数据集中与用户 v 相似的用户数目.

最终, 用户 v 的关系相似度权重表示为:

$$S(v) = \{(u_1, w_1), (u_2, w_2), \dots, (u_n, w_n)\}.$$

3.2 构建用户影响力矩阵

3.2.1 影响力消退函数

在传统的基于交互行为(点赞、转发和评论)的微博影响力模型中, 没有考虑交互行为的时效性因素, 将不同时间发生的交互行为同等对待. 一般来说, 近期交互行为在计算用户影响力时有着更大的贡献. 这是由于随着时间的变化, 用户兴趣会发生偏移, 近期的交互行为更能反映当前的兴趣偏好.

因此, 本文引入遗忘函数来差异性地对待不同时间的交互行为. 定义遗忘函数如下:

$$F(t) = \exp\left(-\frac{\ln 2 \times (t - t_0)}{hl}\right) \quad (7)$$

式中, t 是当前的时间; t_0 是模型建立时的时间; hl 是影响力的半衰期, 是影响力衰退到初始值一半时所用的时间. 当 $t - t_0 = hl$ 时, $F(t) = 1/2$.

3.2.2 短期影响力评估模型

在用户交互模型中, 包含着四个关键要素: 发布者、读者、内容以及读者对内容的反馈. 影响力体现在四者的交互中, 发布者是信息源, 也是影响力的传播源, 读者是信息的接受者, 也就是影响力的接受者. 为了能够快速分析任意时刻用户受到的影响力, 本文建立了短期影响力评估模型(Short TremInfluenceModel, STIM).

短期影响力模型指定了一个大小为 S 的时间窗

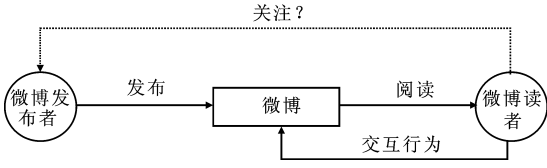


图2 微博用户交互模型

口,覆盖了用于建模的用户交互行为.为了确定时间窗口的大小,本文定义了四个常量:MinDays, MaxDays, MinActionNums, MaxActionNums, 并采用如下策略计算 S :

$$S = \begin{cases} \text{MinDays} & , \text{RAN} \geq \text{MaxActionNums} \\ \text{MaxDays} & , \text{RAN} \geq \text{MinActionNums} \\ 0 & , \text{RAN} < \text{MinActionNums} \end{cases} \quad (8)$$

式中, RAN 是近期交互行为数目.

建立的短期影响力如下所示:

$$\text{Action}(u, v, s, i) = \alpha_1 \text{Like}(u, v, s, i) + \alpha_2 \text{Retweet}(u, v, s, i) + \alpha_3 \text{Comment}(u, v, s, i) \quad (9)$$

$$\text{Influence}(u, v, s, t) = \frac{\sum_{i=1}^m \text{Action}(u, v, s, i)}{3 \times m} \times F(t) \quad (10)$$

式中, α_1, α_2 和 α_3 是三种影响力的权重系数; m 是 u 在大小为 S 的时间窗口内所发微博的总数; $\text{Action}(u, v, s, i)$ 是 v 对 u 的第 i 条微博的交互行为; t 是模型建立时的时间.

u 对 v 的影响力权重如公式(11)所示:

$$w_u^s = \frac{\text{Influence}(u, v, s, t)}{\sum_{i=1}^n \text{Influence}(u_i, v, s, t)} \quad (11)$$

式中, u_i 表示对 v 产生影响的用户; n 是数据集中对 v 产生影响的用户数目.

基于以上分析更新短期影响力评估模型如下:

$$I_v^s = \{(u_1, w_{u_1}^s, T_s), (u_2, w_{u_2}^s, T_s), \dots, (u_n, w_{u_n}^s, T_s)\}.$$

3.2.3 长期影响力评估模型

在长期影响力评估模型中(Long Term Influence Model, LTIM),为了模拟影响力的变化过程,当有新的交互行为发生时,系统会为这些交互行为生成新的影响力模型向量,并更新初始的长期影响力模型,更新如公式(12)所示:

$$I_{nl} = I_{\alpha} F_d \times \frac{N-1}{N} + I_n \times \frac{1}{N} \quad (12)$$

式中, I_{nl} 表示更新后的长期影响力模型; I_{α} 是更新前的长期影响力模型; I_n 是基于最新的交互行为构

建的影响力向量; N 是模型中总的交互行为数目.不同于短期影响力模型,长期影响力模型中遗忘函数半衰期不再是一个常量,会随着时间增长而变长.基于以上分析,构造了新的遗忘函数如下:

$$F(t) = \exp\left(-\frac{\ln 2 \times (t - t_0)}{hl_0 + d_{acc} \times dw}\right) \quad (13)$$

式中, hl_0 是半衰期的初始值; d_{acc} 表示模型建立后到当前所经历的时间.常量 dw 表示 d_{acc} 对遗忘速度的影响程度,通过控制 d_{acc} 和 dw 来控制长期影响力模型的衰落速度.最终长期影响力模型表示为: $I_v^L = \{(u_1, w_{u_1}^L, T_L), (u_2, w_{u_2}^L, T_L), \dots, (u_n, w_{u_n}^L, T_L)\}.$

3.3 基于联合概率矩阵分解的微博关注推荐模型

本文提出的基于联合概率矩阵分解的微博关注推荐模型如下图所示:

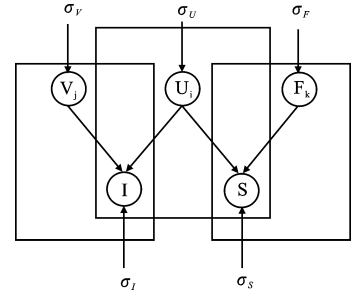


图3 UPMFF 图模型

在短期影响力模型中, I 为 I^s , 基于短期影响力的联合概率矩阵分解模型为 UPMFF-S; 在长期影响力模型中, I 为 I^L , 基于长期影响力的联合概率矩阵分解模型为 UPMFF-L.

UPMFF 模型基于以下假设:

假设 U_i, V_j 和 F_k 的先验概率均服从高斯分布且相互独立, 即:

$$p(U | \sigma_U^2) = \prod_{i=1}^n N(U_i | 0, \sigma_U^2 E) \quad (14)$$

$$p(V | \sigma_V^2) = \prod_{j=1}^n N(V_j | 0, \sigma_V^2 E) \quad (15)$$

$$p(F | \sigma_F^2) = \prod_{k=1}^n N(F_k | 0, \sigma_F^2 E) \quad (16)$$

式中, $N(x | \mu, \sigma^2)$ 表示均值为 μ , 方差为 σ^2 的正态分布的概率密度函数, E 为单位矩阵.

在给定用户 u_i 和 v_j 的潜在特征向量 U_i 和 V_j 后, 用户 u_i 对 v_j 的影响力 I_{ij} 满足均值为 $g(U_i^T V_j)$ 、方差为 σ_i^2 的正态分布且相互独立. 影响力矩阵 I 的条件概率分布如下所示:

$$p(I | U, V, \sigma_i^2) = \prod_{i=1}^n \prod_{j=1}^n [N(I_{ij} | g(U_i^T V_j), \sigma_i^2)]^{H_{ij}^L}$$

(17)

式中, H_{ij}^l 是指示函数:若用户 u_i 对用户 v_j 的行为产生影响,则 $H_{ij}^l = 1$, 否则 $H_{ij}^l = 0$; $g(x) = 1/(1 + e^{-x})$ 是逻辑斯蒂函数, 用于将 $U_i^T V_j$ 的值映射到 $[0, 1]$.

同理, 在确定 U_i 和 F_k 后, 关系相似度矩阵 S 的条件概率如下式所示:

$$p(S | U, F, \sigma_S^2) = \prod_{i=1}^n \prod_{k=1}^n [N(S_{ik} | g(U_i^T F_k), \sigma_S^2)]^{H_{ij}^S} \quad (18)$$

使用贝叶斯公式可推出 U, V, F 的后验概率密度分布函数:

$$\begin{aligned} \ln p(U, V, F | I, S, \sigma_I^2, \sigma_S^2, \sigma_U^2, \sigma_V^2, \sigma_F^2) = & -\frac{1}{2\sigma_I^2} \sum_{i=1}^n \sum_{j=1}^n H_{ij}^l (I_{ij} - g(U_i^T V_j))^2 - \frac{1}{2\sigma_S^2} \sum_{i=1}^n \sum_{k=1}^n H_{ij}^S (S_{ik} - \\ & g(U_i^T F_k))^2 - \frac{1}{2\sigma_U^2} \sum_{i=1}^n U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^n V_j^T V_j - \\ & \frac{1}{2\sigma_F^2} \sum_{k=1}^n F_k^T F_k - \sum_{i=1}^n \sum_{j=1}^n H_{ij}^l \ln \sigma_I^2 - \sum_{i=1}^n \sum_{k=1}^n H_{ik}^S \ln \sigma_S^2 - \\ & d \sum_{i=1}^n \ln \sigma_U - d \sum_{j=1}^n \ln \sigma_V - d \sum_{k=1}^n \ln \sigma_F + C \end{aligned} \quad (19)$$

式中, d 是特征向量的维数; C 是常量; 最大化 $\ln p(U, V, F | I, S, \sigma_I^2, \sigma_S^2, \sigma_U^2, \sigma_V^2, \sigma_F^2)$ 等价于最小化 Q . $Q(U, V, F, I, S) =$

$$\begin{aligned} & \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n H_{ij}^l (I_{ij} - g(V_j))^2 + \frac{\lambda_S}{2} \sum_{i=1}^n \sum_{k=1}^n H_{ij}^S (S_{ik} - \\ & g(U_i^T F_k))^2 + \frac{\lambda_U}{2} \|U\|_F^2 + \frac{\lambda_V}{2} \|V\|_F^2 + \frac{\lambda_F}{2} \|F\|_F^2 \end{aligned} \quad (20)$$

式中, $\lambda_S = \frac{\sigma_I^2}{\sigma_S^2}$, $\lambda_U = \frac{\sigma_I^2}{\sigma_U^2}$, $\lambda_V = \frac{\sigma_I^2}{\sigma_V^2}$, $\lambda_F = \frac{\sigma_I^2}{\sigma_F^2}$, $\|\cdot\|_F$ 是范数. 式(20)可由梯度下降法求得:

$$\frac{\partial Q}{\partial U_i} = \sum_{j=1}^n H_{ij}^l g'(U_i^T V_j) (g(U_i^T V_j) - I_{ij}) V_j + \lambda_S \sum_{k=1}^n H_{ik}^S g'(U_i^T F_k) (g(U_i^T F_k) - S_{ik}) F_k + \lambda_U U_i \quad (21)$$

$$\frac{\partial Q}{\partial V_j} = \sum_{i=1}^n H_{ij}^l g'(U_i^T V_j) (g(U_i^T V_j) - I_{ij}) U_i + \lambda_V V_j \quad (22)$$

$$\frac{\partial Q}{\partial F_k} = \lambda_S \sum_{i=1}^n H_{ik}^S g'(U_i^T F_k) (g(U_i^T F_k) - S_{ik}) U_i + \lambda_F F_k \quad (23)$$

3.4 时间复杂度分析

算法的主要计算开销来自关系矩阵、影响力矩阵、目标函数 Q 和目标函数所对应的梯度下降公式.

在计算关系矩阵时, 假设每个用户的邻居集合为 m_1 , 则对应的关系矩阵的计算时间复杂度为 $O(m_1 n^2)$; 同样, 对于每个用户, 假设其交互行为的集合大小为 m_2 , 影响力矩阵的计算复杂度为 $O(m_2 n^2)$. 公式(20)中目标函数 Q 的时间复杂度是 $O(n_l l + n_s l)$, n_l, n_s 分别是 I 和 S 中非零元素个数. 同理可计算(21) ~ (23) 的时间复杂度为 $O(n_l l + n_s l)$, $O(n_l l)$ 和 $O(n_s l)$. 因此, 每次迭代的总时间复杂度为 $O(n_l l + n_s l)$.

4 实验及结果分析

4.1 实验数据

由于目前缺少公开的中文微博数据集^[6]. 因此本文使用 scrapy 爬虫从新浪微博爬取了五个月(2018.1~2018.5)的微博数据, 获取方式参考文献[7].

4.2 实验评估指标

为了验证本文算法的推荐准确率, 本文首先对基于短期影响力的 UPMFF-S 与基于长期影响力的 UPMFF-L 进行对比实验. 对于短期影响力, 参数分别设置为 $hl=20$, $MinDays=10$, $MaxDays=20$, $MinActionNums=15$, $MaxActionNums=25$; 对于长期影响力, 参数分别设置为 $hl_0=10$, $dw=0.5$; 基于两种影响力模型的 UPMFF-S 和 UPMFF-L 对比如图 4 所示:

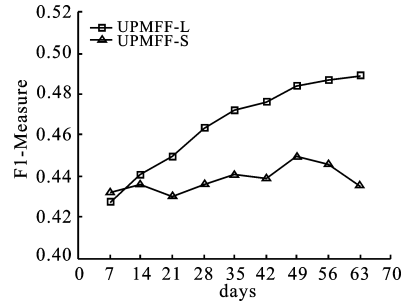


图 4 UPMFF-S 与 UPMFF-L 准确率对比

从图 4 可以看出, 伴随时间的变化, UPMFF-L 推荐准确率先不断升高, 后趋于平缓; UPMFF-S 则处于波动状态. 这是因为 UPMFF-S 模型只考虑了时间窗口内的数据, 在对应时间窗口内, 推荐准确率与窗口内的数据密度有关. 而随着时间的变化, 数据不断增多, UPMFF-L 会不断进行修正, 推荐准确率也就随之不断升高.

基于以上分析, 本文使用基于长期影响力模型的推荐算法 UPMFF-L 与 SoRec^[8]、非负矩阵分解

算法(NMF)^[9]、概率矩阵分解算法(PMF)^[10]进行对比实验. 为了保证实验结果的准确性, 每组实验均重复进行了 10 次, 最终的实验结果取 10 次实验的平均值^[11]. 同时, 经过多次实验测试, 发现当参数 $\lambda_U = \lambda_V = \lambda_F = 0.001$, $\lambda_S = 0.5$, 特征维数 $d = 20$ 时, 算法取得最好的推荐效果. 由不同数据密度下.

实验结果可以看出, 随着推荐数目的不断增多, 本文提出的 UPMFF-L 算法与其余三种推荐算法 SoRec、NMF 和 PMF 的准确率先是不断上升, 最后不断降低, 这是因为随着推荐列表长度的增加, 查准率不断降低, 导致准确率呈现上图中的变化趋势. 数据密度从 80% 下降到 50% 时, UPMFF-L 算法准确率下降 13.7%, SoRec 准确率下降 21.5%, NMF 下降 22.5%, PMF 下降 23.1%. 表明 UPMFF-L 算法对数据密度的变化相对不敏感, 仍能保持较好的推荐准确率.

5 结束语

提出了基于短期影响力模型的 UPMFF-S 算法和基于长期影响力模型的 UPMFF-L 算法. 实验结果表明, UPMFF-L 算法推荐准确率高于 SoRec、PMF 和 NMF 算法. 此外, UPMFF-L 算法在不同数据密度下的实验结果也证明了其可以缓解数据稀疏对推荐准确率的影响.

参考文献:

- [1] 郭磊, 马军, 陈竹敏, 等. 一种结合推荐对象间关联关系的社会化推荐算法[J]. 计算机学报, 2014, 37(1): 219-228.
- [2] TANG J, WANG S, HU X, et al. Recommendation

with Social Dimensions[C]// Thirtieth Aaai Conference on Artificial Intelligence. AAAI Press, 2016.

- [3] 陈婷, 朱青, 周梦溪, 等. 社交网络环境下基于信任的推荐算法[J]. 软件学报, 2017, 28(3): 721-731.
- [4] 徐建民, 刘明艳, 王苗. 基于用户扩展兴趣的微博推荐方法[J/OL]. 计算机应用研究, 2019(06): 1-6[2018-11-09].
- [5] YU Y, MO L, ZHOU J. Social Friend Interest Similarity in Microblog and Its Implication[J]. International Journal of Control & Automation, 2015, 8(11): 21-32.
- [6] 徐志明, 李栋, 刘挺, 等. 微博用户的相似性度量及其应用[J]. 计算机学报, 2014, 37(1): 207-218.
- [7] 穆俊. 基于关联规则挖掘的社区网络数据爬虫算法[J]. 微电子学与计算机, 2018, 35(8): 105-108.
- [8] MA H, YANG H, LYU M R, et al. SoRec: social recommendation using probabilistic matrix factorization[C]// ACM, 2008: 931-940.
- [9] LEE D D, SEUNG H S. Algorithms for non-negative matrix factorization[C]// Advances in neural information processing systems. 2001: 556-562.
- [10] MNIH A, SALAKHUTDINOV R R. Probabilistic matrix factorization[C]// Advances in neural information processing systems. 2008: 1257-1264.
- [11] 孙光福, 吴乐, 刘淇, 等. 基于时序行为的协同过滤推荐算法[J]. 软件学报, 2013, 24(11): 2721-2733.

作者简介:

熊孝全 男, (1993-), 硕士研究生. 研究方向为数据挖掘.
E-mail: xiaoxiongyouxiango@163.com.

张明新 男, (1962-), 博士, 教授、硕士生导师. 研究方向为云计算、数据挖掘、智能控制等.