文章编号：2096-1472(2017)-03-14-03

基于协同过滤的个性化微博推荐算法研究

秦晓晖
（太原工业学院计算机工程系，山西 太原 030008）

摘　要：当前，微博已经成长为世界上最有影响力的社交网络服务之一。随着微博的流行，微博上大量的数据也使得用户无法快速获取他感兴趣的信息。推荐系统是通过研究用户已有数据来发掘用户兴趣，从而为用户推荐可能感兴趣的对象，如产品、网页、微博等。本文介绍了一种基于协同过滤推荐技术的微博推荐算法，从影响用户兴趣度的隐性因素，以及微博互联网中的数据采集和预处理等角度对微博推荐进行研究。使用矩阵分解对隐性因素建模，在已有用户与微博、用户与微博发布者影响因素的基础上，提出微博与微博发布者影响因素，提高了原算法的准确度。

关键词：微博推荐；协同过滤；矩阵分解
中图分类号：TP391　　**文献标识码**：A

A Personalized Micro-Blog Recommendation Algorithm Based on Collaborative Filtering

QIN Xiaohui
(*School of Computer Engineer,Taiyuan Institute of Technology,Taiyuan* 030008,*China*)

Abstract: Currently,micro-blog has become one of the most influential networking services throughout the world. Along with its increasing growth of popularity,the large number of information available on micro-blog has obstructed people from accessing the messages they are interested in.The micro-blog recommendation system picks out and recommends the objects (e.g.products,webpages,micro-blogs,etc.) via analyzing the existing data of the user.The paper proposes a micro-blog recommendation algorithm based on the collaborative filtering technique,explores some recessive factors which may influence user's interest and studies micro-blog recommendation from the perspective of data collecting and preprocessing on micro-blog networks.While the previous studies only focus on the relationship between the user and the publisher,and that between the user and the micro-blog post,this paper adopts matrix decomposition to model recessive factors and proposes the influence factors between the publisher and the micro-blog post.Finally,the experimental results show that the new algorithm significantly improves the accuracy of micro-blog recommendation.

Keywords:micro-blog recommendation;collaborative filtering;matrix decomposition

1 引言(Introduction)

目前被广泛应用的协同过滤算法[1]在推荐系统[2]中发挥着很重要的作用。随着信息种类的丰富，我们需要对一些很难基于内容来分析的信息，尤其是对一些复杂的甚至难以表达的概念进行兴趣分析，协同过滤算法表现出了一定的优越性。矩阵分解算法[3]目前已经被广泛地应用于推荐系统中，它作为隐语义模型中的一种方法取得了一定的成就。协同过滤算法一般可以分为基于相似邻居的方法[4,5]和基于模型的方法[6,7]这两大类，目前隐因子概率模型或者矩阵分解模型经常被用来解决一些问题。本文主要使用基于模型算法中的矩阵分解算法，具体使用隐因子模型来度量影响微博用户喜好的一些隐性因素。

本文向用户进行微博推荐是通过用户对微博的兴趣度来分析的，那么就需要找出影响用户对于微博兴趣度的一些隐性因素，而矩阵分解作为一种隐含语义模型可以很好地帮我们找出这些隐性因素。因此在微博中并不需要指出微博具体的属性类别，可以使用隐语义模型构建矩阵：比如构建一个user−tweet矩阵R见公式(1)，其中R_{ij}表示用户i对微博j的兴趣度，通过对矩阵R分解得到矩阵P和矩阵Q，其中f为影响用户兴趣度的隐性属性，这个过程就称为奇异值分解[7,8]。

$$\begin{matrix} R & tweet\,1 & tweet\,2 \\ user\,1 & R11 & R12 \\ user\,2 & R21 & R22 \end{matrix} = \begin{matrix} P & f1 & f2 \\ user\,1 & P11 & P12 \\ user\,2 & P21 & P22 \end{matrix} \times \begin{matrix} Q & tweet\,1 & tweet\,2 \\ f1 & Q11 & Q12 \\ f2 & Q21 & Q22 \end{matrix} \quad (1)$$

通过矩阵P和矩阵Q想乘即可得出R中缺失的兴趣度，详细求解见公式(2)：

$$R_{ij} = P_i Q_j = \sum_{k=1}^{K} P_{i,k} Q_{k,j} \quad (2)$$

从上述过程可以看出我们无需确定属性的具体类别和属性的个数，只需要设置隐因子模型中的属性个数值作为属性分类的粒度即可，值越大即代表分类的粒度越细。通过隐因子模型，在不知道微博的类型和用户喜欢的微博类别的前提下也可以得到用户对每个类别的兴趣度。

2 基于协同排序的微博推荐算法(Collaborative ranking method for tweet recommendation)

2.1 微博排序优化准则

本文研究用户对微博喜好度的排序，我们使用协同排序算法，它是基于隐因子模型的协同过滤方法。首先定义表示R^d低维向量，同时定义$p_u \in R^d$和$q_i \in R^d$来表示用户和微博的属性空间向量。那么就可以通过公式(3)来预测用户u对微博i的喜好度：

$$y_{u,i} = {p_u}^T q_i \tag{3}$$

由于我们最终要获得的是用户对微博兴趣度的排序结果，而预测值y为用户u对微博i喜好度，这里我们认为用户对转发过的微博喜好度大于未转发过的微博即$y_{u,k} > y_{u,h}$，为了实现微博的排序，我们将目标函数定义为计算求微博k在喜好度排序中比微博h靠前的可能性，详见公式(4)：

$$P(r(k) > r(h)|u) = \frac{1}{1+\mathrm{e}^{-(y_{u,k}-y_{u,h})}} \tag{4}$$

其中，k为转发过的微博，h为非转发的微博，r表示微博排序，P表示微博k比微博h的排序靠前的概率，等式右边是对$y_{u,k} > y_{u,h}$的归一化处理。这里，我们构建表示用户对转发微博喜好度大于非转发微博喜好度的数组D，见公式(5)：

$$D = \{< u,k,h > | k \in Re(u), h \notin Re(u)\} \tag{5}$$

其中，*Re(u)*为用户u转发微博的集合。依据前面的假设，我们认为所有用户对转发微博的喜好度比非转发微博高，因此Re集合里的所有微博k都能与非Re集合里的微博h组成D中的一个元素<*u,k,h*>，这里我们定义k为正例，h为负例。对公式(6)取对数进行最大似然估计更加便于计算，最终转化为求解目标函数(6)：

$$\mathrm{Min} \sum_{<u,k,h>\in D} \ln(1+\mathrm{e}^{-(y_{u,k}-y_{u,h})}) + r \tag{6}$$

其中，r为正则化参数。

2.2 基于矩阵的隐因子分解模型

本文中通过研究用户、微博和微博发布者三者之间的隐性因素来预测用户对微博的兴趣度。因此可以将用户—微博矩阵使用SVD方法拆分为三个矩阵，具体分解为用户—微博矩阵、用户—发布者矩阵、发布者—微博矩阵，矩阵分解的过程不仅极大地丰富了我们的模型，使得一些潜在影响因素被挖掘出来，而且一定程度上缓解了由于转发行为少而导致的矩阵稀疏问题。

(1)用户—微博主题偏好分解

由于用户微博转发次数导致数据稀疏的问题，本文通过微博内容信息来缓解该问题，不同的主题可以使用不同的词来代表，因此可以将微博的隐因子模型转化为主题词语的隐因子组合，于是转化为分解模型(7)：

$$y_{u,i} = {p_u}^T (\frac{1}{Z}\sum_{w\in T_i} q_w) \tag{7}$$

其中，p_u表示用户—属性矩阵，q_w表示词—属性矩阵，q_w矩阵中的每一个词w都属于微博i，Z为微博I中词的个数，乘以$\frac{1}{Z}$对每个词的权重进行归一化。这样的转化由原来的用户对一条微博的喜好度转变为用户对词或主题的喜好度，从而缓解了矩阵稀疏问题。

(2)用户—发布者社会关系分解

除了微博内容还可以将用户与发布者的社会关系也考虑进模型。如果用户对发布者发布的微博主题感兴趣的话，也就是用户的兴趣与该微博发布者的微博主题很相似，那么该用户转发该发布者的微博的可能性就比较高，因此通过用户与微博发布者之间的隐性因子可以预测用户转发该条微博的概率，详见公式(8)：

$$y_{u,i} = {p_u}^T d_{p(i)} \tag{8}$$

其中，$d_{p(i)}$表示发布者p发布的微博i的发布者隐性因子矩阵。这种分解不考虑微博的内容计算转发一条微博的先验概率。考虑社交关系进我们的模型通过线性组合可以得到公式(9)：

$$y_{u,i} = {p_u}^T (\frac{1}{Z}\sum_{w\in T_i} q_w + \alpha d_{p(i)}) \tag{9}$$

(3)发布者—微博主题权威性分解

在以上分析的基础上我们又考虑了发布者与微博主题权威性之间的隐性因子对用户兴趣度的影响。这里提出的微博权威性对用户微博转发行为的影响不是基于用户来考虑的，与以上两种分析是不同的。通常如果一些权威专家发布一些他所在的专家领域相关的微博，那么这种微博话题通常会比较吸引用户的注意力，用户会倾向于转发此类微博。计算微博权威性隐性因子详见公式(10)：

$$y_{u,i} = \frac{1}{Z}\sum_{w\in T_i} {q_w}^T d_{p(i)} \tag{10}$$

通过线性组合将微博权威性的隐性因子考虑进我们的模型可以转化为公式(11)：

$$y_{u,i} = {p_u}^T(\frac{1}{Z}\sum_{w\in T_i} q_w + \alpha d_{p(i)}) + \frac{1}{Z}\sum_{w\in T_i} {q_w}^T \beta d_{p(i)} \quad (11)$$

公式(11)表示通过挖掘用户、微博和发布者这三者中的两两之间的隐性因子度量用户的兴趣度，不仅全面地考虑了多种隐性因子丰富了模型，而且一定程度上缓解了数据稀疏的问题。

(4)参数估计

本文使用线性加权的方法来预测用户对微博的兴趣度，其中 α 为发布者对微博影响因子的权重，β 为发布者对微博主题影响因子的权重。2.1节中给出的目标函数(6)是求解的对象，本文中使用梯度下降的方法得到最优解即对目标函数求导。首先对矩阵进行初始化，这里我们使用随机数，然后通过对构造的数据集D中的每一组元素计算梯度来不断更新矩阵中的值直到循环终止得到最优解。其中，梯度更新系数详见公式(12)到公式(17)：

$$-\frac{\partial L}{\partial p_u} = \hat{e}(\frac{1}{Zs^+}\sum_{s\in k} q_s - \frac{1}{Zs^-}\sum_{s\in h} q_s + (d_{p(k)} - d_{p(h)})) - \sigma_1 p_u \quad (12)$$

$$-\frac{\partial L}{\partial {q_w}^+} = \hat{e}\left(\frac{1}{Zj^+}p_u + d_{p(k)}\right) - \sigma_2 {q_w}^+ \quad (13)$$

$$-\frac{\partial L}{\partial {q_w}^-} = \hat{e}\left(\frac{1}{Zj^-}p_u + d_{p(k)}\right) - \sigma_2 {q_w}^- \quad (14)$$

$$-\frac{\partial L}{\partial d_{p(k)}} = \hat{e}\left(p_u + \frac{1}{Zj^+}{q_w}^+\right) - \sigma_3 d_{p(k)} \quad (15)$$

$$-\frac{\partial L}{\partial d_{p(h)}} = \hat{e}\left(p_u + \frac{1}{Zj^-}{q_w}^-\right) - \sigma_3 d_{p(h)} \quad (16)$$

$$-\frac{\partial L}{\partial b_j} = \hat{e}({\gamma_j}^+ + {\gamma_j}^-) - \sigma_4 b_j \quad (17)$$

其中，“+”表示在数据集中转发微博中的值，“–”表示在非转发微博中的值，$\hat{e}$表示真实值与预测值之间偏差的概率，详见公式(18)：

$$\hat{e} = 1 - P(r(k) > r(\hat{k})|u) = 1 - \frac{1}{1+e^{-(y_{u,k}-y_{u,h})}} \quad (18)$$

算法中不停循环使得模型中的权重值不断更新，向着梯度下降的方向直到循环终止得到最优解。

3 实验(Experiment)

3.1 数据来源

本文根据特定的需求在新浪微博使用爬虫系统[9]获取相关数据，网络爬虫作为搜索引擎的核心技术是一种自动提取网页信息的计算机程序或者自动化脚本[10]。本文的实验数据通过随机选取一个微博用户，然后以发射状不断爬取该用户的关注者的数据，以及关注者的关注者的数据，从爬取的数据中找出1024个关注者人数超过15的微博用户的主页信息作为实验数据。

3.2 评价标准

考虑到推荐结果中成功率的问题，本文使用平均准确率来评价预测结果的准确度。模型的推荐结果是微博排序，同时还可以用准确度关联成功推荐的微博的排序位置从而使得推荐模型得到更准确的评估，即成功推荐的微博排序越靠前，那么平均准确率越高。如果系统没有成功推荐的微博，那么准确率记为0。评估公式详见(19)：

$$AP = \frac{\sum_{n=1}^{N} P@n \times retweet(n)}{|R|} \quad (19)$$

其中，N为测试集中的微博数量，R为测试集中用户转发过的微博数量，*retweet(n)*为布朗函数，当第n条微博是用户转发过的微博(即成功推荐)，*retweet(n)*的值为1，当第n条微博为用户没有转发过的微博时，*retweet(n)*的值为0。*P@n*为取排序结果中前n条微博时的准确度。当计算出所有用户的AP值时就可以得到MAP的值，详见公式(20)，其中N为用户总数。

$$MAP = \frac{AP}{N} \quad (20)$$

3.3 实验结果

本文通过与其他几种方法的对比实验结果来验证算法的有效性。按照时间排序的方法是指所有微博按照时间排序不通过其他算法重排序，这种方法表现微博最直接、最原始的状态，但却忽略了用户兴趣对微博排序的影响，与这种方法得到的结果相对比将有效地说明本文中算法研究的意义和必要性。按相似度排序的方法是按照微博与用户标签的相似性来排序的，这里使用余弦相似度来计算相似度，标签是指用户历史微博和转发微博历史里面的关键词的集合。原始[11]方法在隐性因素方面只考虑主题层次和社会关系层次。矩阵分解模型算法SVD在原始算法的基础上添加影响用户兴趣度的微博权威性隐性因素预测用户兴趣度。该算法也使用随机梯度算法来估计实验参数，实验中矩阵分解过程中使用到的K值取30准确率最高。

表1 所有方法的MAP值

Tab.1 MAP results of all methods

方法	时间排序	相似度	原始方法	SVD
MAP	0.158	0.102	0.151	0.304

4 结论(Conclusion)

按照时间序列排序的推荐方法依赖于用户的登录时间，用户对登录时间前后的微博转发概率大，因此预测准确度很低。按照相似度的排序只通过关键词计算微博表面相似度，忽略了内在语义。原始方法没有考虑微博与微博发布者之间的隐性因素而低于SVD方法。

参考文献(References)

[1] Shi Y,Larson M,Hanjalic A.Collaborative Filtering Beyond the

User-Item Matrix:A Survey of the State of the Art and Future Challenges[J].ACM Computing Surveys (CSUR),2014,47(1):3.

[2] Yang X,et al.A Survey of Collaborative Filtering Based Social Recommender Systems[J].Computer Communications, 2014,41:1-10.

[3] Levy O,Goldberg Y.Neural Word Embedding as Implicit Matrix Factorization[C].Advances in Neural Information Processing Systems,2014:2177-2185.

[4] Sarwar B.,et al.Item-Based Collaborative Filtering Recommendation Algorithms[A].Hypermedia Track of the 10th International World Wide Web Conference,2001:285-295.

[5] Shi Y.,Larson M.,Hanjalic A.Exploiting User Similarity Based on Rated-Item Pools for Improved User-Based Collaborative Filtering[A].Third ACM Conference on Recommender Systems,2009:125-132.

[6] Koren Y.Factorization Meets the Neighborhood:a Multifaceted Collaborative Filtering Model[A].The 14th ACM SIGKDD International Conference on Knowledge,2008:426-434.

[7] Rendle S.The IEEE International Conference on Data Mining[C].Factorization machines,2010:995-1000.

[8] Cao Y.,et al.Adapting Ranking SVM to Document Retrieval[C].The 29th Annual International SIGIR Conference,2006:186-193.

[9] 孙立伟,何国辉,吴礼发.网络爬虫技术的研究[J].电脑知识与技术,2010,6(15):4112-4115.

[10] 高建煌.个性化推荐系统技术与成用[D].中国科学技术大学,2010.

[11] Chen K.,et al.Collaborative Personalized Tweet Recommendation[A].The 35th International ACM SIGIR Conference on Research and Development in Information Retrieval,2012:661-670.

作者简介：

秦晓晖(1987-)，女，硕士，助教.研究领域：中文信息处理，人工智能.

(上接第25页)

数量，然后就可以确定母鸡b的数量(b=100-a-c)；当然，我们也可以先确定母鸡b和小鸡c的数量，再确定公鸡a的数量，此时所使用的二重循环语句是：

```
for(b=1;b<=25;b++)
for(c=63;c<=87;c+=3)
{a=100-b-c;
if(5*a+3*b+c/3==100)
    printf("公鸡=%d,母鸡=%d,小鸡=%d\n",a,b,c);}
```

也可以先确定公鸡a和母鸡b的数量，再确定小鸡c的数量，此时所使用的二重循环语句是：

```
for(a=1;a<=14;a++)
for(b=1;b<=25;b++)
{c=100-a-b;
if((5*a+3*b+c/3==100)&&(c%3==0))
    printf("公鸡=%d,母鸡=%d,小鸡=%d\n",a,b,c);}
```

根据对算法五的三种情况进行对比可以发现，情况一的执行次数为126，情况二的执行次数为25*9=225，情况三的执行次数为14*25=350，显然选择取值范围小的两个变量作为循环变量来构造二重循环是比较合理的，当然这三种情况的算法执行效率都要优于前面的算法。

5 结论(Conclusion)

以上五个算法是应用多重循环语句对“百钱买百鸡”问题的算法分析，由差到优循序渐进地对算法进行了改进，通过每一次的改进降低了算法的执行时间，从最初的10^6次的循环执行次数降到了最后的126次，最终得到了最为理想的算法。所以，我们在进行算法设计时，不应该只是得出了正确的算法就可以了，而是要尽量去寻找最优的算法，要具有不断地对已有算法设计进行改进和优化的精神。当然，该问题的解决方法不止于此，必定还会有一些更优的算法值得我们去探索。

参考文献(References)

[1] Fathima H.,Musthafa A.Syed.Optimization Based Routing Algorithms[J].International Journal of Applied Research on Information Technology and Computing,2014,5(1):55-70.

[2] Guang-Yu Zhu,Wei-BoZhang.Optimal Foraging Algorithm for Global Optimization[J].Applied Soft Computing,2017,51:294-313.

[3] R.VenkataRao,G.G.Waghmare.A New Optimization Algorithm for Solving Complex Constrained Design Optimization Problems[J].Engineering Optimization,2017,49(1):60-83.

[4] 黄隆华,陈志辉.算法设计与分析课程的“百钱买百鸡问题”趣用[J].计算机教育,2016(3):143-145.

[5] 耿国华.算法设计与分析[M].北京:高等教育出版社,2012(1):20-22.

[6] 许桂平.浅析C语言三种循环结构语句[J].考试周刊,2014(21):117-118.

[7] 任爱华.C语言程序设计[M].北京:中央广播电视大学出版社,2015:66-95.

[8] 马学敏.计算机C语言循环语句的应用研究[J].中国新通信,2016(17):87-88.

作者简介：

龙敏敏(1979-)，女，本科，讲师.研究领域：计算机应用技术，计算机教育教学.