基于偏见修正的联合矩阵分解算法

李 铭[1] 岳 宾[2] 代永平[1]

(南开大学电子信息与光学工程学院 天津 300350)[1] (南开大学计算机与控制工程学院 天津 300350)[2]

摘 要 目前协同过滤的主流方法是矩阵分解模型。针对传统矩阵分解方法没有考虑用户偏见和物品隐含特征对推荐质量的共同影响,在矩阵分解模型的基础上提出了一种基于用户偏见修正的联合矩阵分解算法(联合分解物品评分矩阵和物品共现矩阵)。在不同基准数据集上的实验结果反映了所提策略的合理性,并通过基于排序的指标证明了所提模型比传统矩阵分解模型在性能上有较大幅度的提升。

关键词 协同过滤,矩阵分解,偏见修正,隐式反馈

中图法分类号 TP391 **文献标识码** A

Collective Matrix Factorization Algorithm Based on Bias Amendment

LI Ming[1] YUE Bin[2] DAI Yong-ping[1]

(College of Electronic and Optical Engineering, Nankai University, Tianjin 300350, China)[1]

(College of Computer and Control Engineering, Nankai University, Tianjin 300350, China)[2]

Abstract Although matrix factorization model has become the major method in the collaborative filtering, it ignores the combined influence of the user bias and latentitems characteristics on recommendation quality. Therefore, this research proposed a collective matrix factorization algorithm, which factorizes items rating matrix and items co-occurrence matrix to amend user bias based on matrix factorization model. The experimental results from different benchmark datasets prove the rationality of the combined factorization algorithm, and indicate greater improvement in the ranking-based metrics in comparison with the traditional matrix factorization model.

Keywords Collaborative filtering, Matrix factorization, Bias amendment, Implicit feedback

作为推荐系统中应用最广泛的推荐算法,协同过滤已经取得了巨大的成功,但其仍然面临很多的挑战,其中很重要的一点便是数据量巨大,而且我们能够利用的有效用户反馈信息极度稀疏。针对上述问题,目前主流的解决方法是矩阵分解,也就是降维的方法。主要的研究包括联合矩阵分解[1]、监督/半监督矩阵分解[2]以及联合概率矩阵分解[3]等。但是上述矩阵分解方法通常将用户是理性的作为前提条件,即用户评分是客观的且用户的评价标准统一。然而,实际情况是每一个用户都会根据自己的判断标准做出评价。例如,存在评分集合(评分范围为1~5之间的整数)$a=\{4,0,5,4,5,4,5,5,0\}$,$b=\{3,0,1,3,1,3,1,1,0\}$,集合b对应用户的评判标准相对严格,而集合a则正好相反。通常按照传统的过滤方法,集合a中除了没有评分的项之外,所有的评分都经过滤而保留,而b集合的评分则都被过滤掉。但是考虑到推荐过程中需要消除不同用户的偏见,更合理的方法是过滤掉集合a中的评分为4的项并保留集合b中的评分为3的项。基于上述分析,本文在加权矩阵分解模型[15]基础上,考虑用户偏见可能造成的影响,提出一种引入用户偏见修正的加权矩阵分解算法(WMFAB)。实验结果表明,此方法可以明显提高推荐的质量。

另外,由于传统矩阵分解模型通常只考虑用户与物品之间的关系,而忽略物品之间的关系,因此利用修正用户偏见的信息构造物品共现矩阵[10],进一步提出联合分解用户偏好矩阵和物品共现矩阵来学习得到用户因子矩阵和物品因子矩阵的方法。提出对已经编码的用户偏好信息(即用户-物品-带偏见的评分三元组集合)修正用户偏见,同时考虑到显式反馈数据收集的局限性,提出专注于隐式反馈的模型(也能扩展到显式反馈数据集),提高了算法的可扩展性。实验结果表明优化的算法进一步提高了系统的推荐质量,同时证明该模型具有为活跃度较低(消费物品数目较少)的用户提供有效推荐的能力。

本文第1节介绍相关的研究;第2节给出模型以及算法描述;第3节通过将算法应用于两个不同的数据集来验证其有效性;最后对全文进行总结与展望。

1 相关研究

协同过滤主要分为基于邻域(Memory)的方法和基于隐语义模型(Model)的方法。矩阵分解作为隐语义模型中最成功的一种实现,受到了研究者的广泛关注,其中包括SVD(奇异值分解)、PMF(概率矩阵分解)、NMF(非负矩阵分解)等方法。独立于用户和物品本身并能够反映它们本质的因素被称为偏置项(可以作为参数学习得到)。其中Koren等人[6]在矩阵分解模型的基础上将评分观测值分为全局平均值、物品偏差、用户偏见以及用户特征因子和物品特征因子二者点积,通

李 铭(1991-),男,硕士生,主要研究方向为数据挖掘,E-mail:2120150242@mail.nankai.edu.cn;岳 宾(1992-),男,硕士生,主要研究方向为数据挖掘;代永平(1968-),男,博士,副教授,主要研究方向为模式识别。

过这 4 部分能够更全面地表达用户对物品的预测。Shi 等人[7]在 PMF 的基础上引入物品偏置和用户偏见(二者都服从均值为零的高斯正态分布),可以进一步提高推荐精度。文献[8]将偏置部分加入到 SVD++(添加隐式反馈因素)和邻域模型中,并得到了更好的推荐效果。基于上述研究,本文对用户偏见建模,在矩阵分解模型中加入偏置项,并同时引入物品共现模型来共同学习用户因子和物品因子。

联合矩阵分解[1,9,16]是指通过联合分解多重相关矩阵来改进预测精度,通常代表多重关系的多重矩阵包含相同的实体,我们可以从包含相同实体的关系矩阵中获取更多的额外信息。一般多重关系学习的一个典型例子是电影评分预测:定义实体包括用户、电影、电影类型、演员,其中可以对实体关系进行建模的包括用户对电影的评分、电影所属的类型以及电影中的演员角色。考虑到尽可能挖掘到更多的信息源(用户对其它物品的偏好、其它用户的偏好以及每一个用户和物品的特征)对于预测用户对某一个物品的偏好是有益的,目前的矩阵分解很多都是通过增加和用户或物品相关的额外有效信息来改进推荐质量。例如,这样的边信息包括基于标签的信息[12]、基于文本的信息(用户评价)[13]以及基于图片的信息[14]。Liang 等人[10]于 2016 年提出了 Cofactor 联合分解模型,在除去基本的用户物品偏好矩阵外,该模型没有引入其他额外信息,其联合分解的物品共现矩阵(物品和上下文物品)受到了词嵌入[5]模型的启发,并由用户历史数据定义得到。而本文在 Cofactor 模型[10]的基础上,进一步在修正用户偏见的评分矩阵上得到了物品共现矩阵,并将其用于联合矩阵分解。

2　联合分解模型

2.1　模型定义

矩阵分解通常给定稀疏用户物品关系矩阵 $P\in R^{U\times I}$(用户 $u=1,\cdots,U$,物品 $i=1,\cdots,I$),定义分解后的低维矩阵中的每一个用户隐式表示为 x_u,每一个物品隐式表示为 y_i,这样根据二者点积,可以得到用户 u 对物品 i 的预测值为 $x_u^{\mathrm{T}}y_i$,并通过优化方法(如交替最小二乘法)来逼近真实值。考虑到收集显式反馈的局限性,HU 等人[15]提出了专门针对隐式反馈的矩阵分解模型,具体的用户和物品隐含表示通过最小化目标损失函数式(1)得到。鉴于本文专注于处理隐式反馈,将文献[15]提出的加权矩阵分解 WMF 模型作为基准。

$$\zeta_{mf}=\sum_{u,i}c_{ui}(p_{ui}-x_u^{\mathrm{T}}y_i)^2+\lambda(\sum_u\|x_u\|^2+\sum_i\|y_i\|^2)\quad(1)$$

本文中的联合分解模型主要由两部分组成:1)用户偏见修正的矩阵分解模型;2)物品共现模型。

传统矩阵分解模型建模的用户是理性的用户,当对数据集中的用户评分进行过滤时(以本文的 Movielens 数据集为例),因为要取得正反馈的效果,所以在将其转换为隐式类型数据的过程中通常只保留 4 以上的评分作为模型的输入。但在实际情况中,考虑到所有用户不可能都是理性用户,更可能的情况是每个特定的用户都会存在个人的偏好。例如,某一个用户对绝大多数的物品评分都为 5,而针对其中某一物品评分为 4 的情况,按照传统模型,这一物品评分应该加入用户-物品关系矩阵,但实际上这一物品评分明显低于该用户的评分均值,不属于正反馈,所以不应该将其加入输入矩阵。同理,某一用户对绝大多数物品的评分都为 1,而对某一物品评分为 3,按照传统模型,这一物品评分不应该选入输入矩阵,然而实际上该物品的评分明显高于该用户的评分均值,属于正反馈,应该选入输入矩阵。鉴于上述情况,我们在模型中利用具体用户的评分均值对用户的偏见进行修正,并通过修正用户偏见的评分矩阵定义物品共现矩阵。

联合分解模型中的物品共现模型与从单词序列中学习到的单词嵌入模型[4-5]类似。物品嵌入[11]与最近流行的词嵌入(word2vec[4])类似,即在 Harris 分布假设(假设在相似上下文下的单词具有相似的含义)的基础上依据一个单词-上下文矩阵来表示单词,矩阵中的每一行对应一个单词,每一列对应一个上下文单词,矩阵中的具体条目 m_{ij} 表示单词和上下文单词之间的联系。式(2)中的 PMI(点互信息)被广泛应用于单词相似度任务中,实际中便是通过它计算。其中文献[4]提到的 Skip-gram with Negative Sampling(SGNS)训练词嵌入模型的方法等价于隐式地分解加权的 Shifted-Pointwise Mutual Information(SPMI)矩阵,式(3)中的 k 是负样本值。Levy 等人[5]在上述模型基础上对式(4)的 Sparse Shifted Positive PMI(SPPMI)矩阵通过降维(例如奇异值分解)来学习得到词嵌入,其中将所有的负值替代为 0,是因为人们很容易想到词之间的正向联系,也就是说两个词的相似度更容易被共有的正向上下文影响,而非负面的上下文。鉴于 SPPMI 矩阵对于负样本取样的优势,本文也选择相同的做法来构造物品共现矩阵。

$$PMI(i,j)=\log\frac{\#(i,j)\cdot D}{\#(i)\cdot\#(j)}\quad(2)$$

$$SPMI(i,j)=PMI(i,j)-\log k\quad(3)$$

$$SPPMI(i,j)=\max\{PMI(i,j)-\log k,0\}\quad(4)$$

我们定义稀疏的用户-物品评分矩阵为 $Y\in R^{U\times I}$,修正用户偏见的关系矩阵为 $P\in R^{U\times I}$,用户消费物品的共现矩阵为 $M\in R_+^{I\times J}$。首先将初始评分矩阵与相应的用户评分均值矩阵相减得到修正后的用户-物品关系矩阵 P,然后通过修正用户偏见的关系矩阵得到物品共现矩阵 M。假定每一个用户的物品消费顺序(时间戳)是不可用的,对于一个特定的用户消费物品 i,定义它的上下文 j 是用户消费历史中的其他所有物品。通过联合分解矩阵 P 和矩阵 M 来得到用户嵌入和物品嵌入,联合分解的结果是用户因子矩阵和物品因子矩阵二者的乘积,分别用 $\theta_u\in R^K$ 和 $\beta_i\in R^K$ 表示。物品共现模型所需要的信息都可以从修正用户偏见的关系矩阵中获得。我们的模型专注于处理隐式反馈数据,同时也能处理显式反馈数据。据上,得出联合分解的目标损失函数:

$$\zeta_{co}=\overbrace{\sum_{u,i}c_{ui}(P_{ui}-b_u-\theta_u^{\mathrm{T}}\beta_i)^2}^{\text{Amend user bias\_MF}}+\overbrace{\sum_{m_{ij}\neq0}(m_{ij}-\beta_i^{\mathrm{T}}\gamma_j-w_i-c_j)^2}^{\text{item embedding}}+\lambda_\theta\sum_u\|\theta_u\|_2^2+\lambda_\beta\sum_i\|\beta_i\|_2^2+\lambda_\gamma\sum_j\|\gamma_j\|_2^2\quad(5)$$

公式中的后 3 项是正则化项(在这里取范数的平方和),是为了防止模型过拟合。把用户偏见 b_u 和上下文嵌入 γ_j 作为附加的模型参数,选择对用户因子、物品因子和上下文因子进行正则化处理,而不对用户偏见 b_u、物品偏差 w_i 和上下文偏差 c_j 进行正则化处理。尺度参数 c_{ui}(见式(6))是一个超参数,它的相对比例将会平衡模型中的矩阵分解和物品嵌入部分,将它的值设置得较大将使得共现矩阵带来的正则化影响变小,反之亦然。在我们的实证研究中,将基于验证集的推荐结果来选择其取值。同样地,正则化参数 λ_θ 和 λ_β 也是从验证数据中选出的超参数。目标中的矩阵分解模型和物品嵌入模型同时包含物品因子 β_i,物品因子 β_i 必须同时考虑用户-物品

关系和物品-物品共现关系，这也是引入物品共现信息的根本原因。可以把上述目标理解为对用户偏见修正的矩阵分解目标进行正则化处理，正则项是物品共现矩阵。

$$c_{ui}=1(1+\alpha P_{ui}) \tag{6}$$

本文所引入的偏好修正矩阵分解模型消除了用户偏见对推荐结果的影响，更重要的是由于共现矩阵中的物品和上下文集合同样来自于修正过的用户物品关系集合，理论上推荐效果能够更好地反映实际反馈情况，实验指标的提高证明了上述方法的合理性。

2.2 推理和优化算法

对多数据输入源的预测模型进行公式化定义。鉴于求解优化的重要性，下面将分别给出具体的推理依据以及详细的算法优化过程。另外，模型求解的目标是为每一个用户和每一个物品找到对应的向量表示，迭代学习的过程等价于最小化损失函数(见式(5))，我们使用交替最小二乘法的优化方法[15]。关于模型参数$\{\theta_{1:U},\beta_{1:I},\gamma_{1:J},w_{1:I},c_{1:J}\}$，取损失函数$\zeta_{\infty}$的梯度，并设置为0，由此得出以下参数更新的推导：

$$\theta_u\leftarrow(\sum_i c_{ui}\beta_i\beta_i^{\mathrm{T}}+\lambda_\theta I_k)^{-1}(\sum_i c_{ui}(p_{ui}-b_u)\beta_i) \tag{7}$$

$$\beta_i\leftarrow(\sum_u c_{ui}\theta_u\theta_u^{\mathrm{T}}+\sum_{j:m_{ij}\neq 0}\gamma_j\gamma_j^{\mathrm{T}}+\lambda_\beta I_K)^{-1}*(\sum_u c_{ui}(p_{ui}-b_u)\theta_u+\sum_{j:m_{ij}\neq 0}(m_{ij}-w_i-c_j)\gamma_j) \tag{8}$$

$$b_u\leftarrow\frac{1}{|\{i:P_{ui}\neq 0\}|}\sum_{j:m_{ij}\neq 0}(P_{ui}-\theta_u^{\mathrm{T}}\beta_i) \tag{9}$$

$$\gamma_j\leftarrow(\sum_{i:m_{ij}\neq 0}\beta_i\beta_i^{\mathrm{T}}+\lambda_\gamma I_K)^{-1}(\sum_{i:m_{ij}\neq 0}(m_{ij}-w_i-c_j)\beta_i) \tag{10}$$

$$w_i\leftarrow\frac{1}{|\{j:m_{ij}\neq 0\}|}\sum_{j:m_{ij}\neq 0}(m_{ij}-\beta_i^{\mathrm{T}}\gamma_j-c_j) \tag{11}$$

$$c_j\leftarrow\frac{1}{|\{i:m_{ij}\neq 0\}|}\sum_{j:m_{ij}\neq 0}(m_{ij}-\beta_i^{\mathrm{T}}\gamma_j-w_i) \tag{12}$$

算法 1 迭代优化算法

```
Input:
  Amend User bias_Matrix:Pui
  Item co-occurrence_Matrix:SPPMI
  Initial θ,β,b,γ,w,c
Output:
  User_factor:θ;Item_factor:β
  Procedure Learning(P,M):
    for A fixed number of iterations do
      for u=1,2,…,U do
      θu←(Σi cui βi βiT+λθ Ik)^-1 (Σi cui pui βi)
      if (θ has been updated) then
        for i=1,2,…,I do
          βi←(Σu cui θu θuT+Σ_{j:mij≠0} γj γjT+λβ IK)^-1 * (Σu cui pui θu+
          Σ_{j:mij≠0}(mij−wi−cj)γj)
      if (θ,β have been updated) then
        for u=1,2,…,U do
          bu←1/|{i:Pui≠0}| Σ_{j:mij≠0}(Pui−θuT βi)
      if (β has been updated) then
        for j=1,2,…,J do
      γj←(Σ_{i:mij≠0} βi βiT+λγ IK)^-1 (Σ_{i:mij≠0}(mij−wi−cj)βi)
      if (θ,β have been updated) then
        for i=1,2,…,I do
          wi←1/|{j:mij≠0}| Σ_{j:mij≠0}(mij−βiT γj−cj)
      if (θ,β,w have been updated) then
        for j=1,2,…,J do
      cj←1/|{i:mij≠0}| Σ_{j:mij≠0}(mij−βiT γj−wi)
      Validate(θ,β,Vad_data):
        Return vad_ndcg
      if self.vad_ndcg > vad_ndcg
        break
      else
        self.vad_ndcg=vad_ndcg
  end procedure
```

该算法可以在用户和物品之间实现并行化计算，能够显著提高模型求解的计算效率，我们把交替最小二乘法的每一次更新看作是一次加权岭回归。与文献[9]的区别主要是更新β_i的方法不同，可以理解为对β_i的更新是在两个数据源中共同执行岭回归，数据源是包含协变量θ_u的偏好修正信息P_{ui}以及包含协变量γ_j的物品共现信息m_{ij}。

3 实验与分析

在实验设计和结果分析部分，根据推导的求解公式实现本文所提出的修正用户偏见的联合矩阵分解算法。下面将首先介绍数据集，然后说明相应的评价标准以及对比算法，最后给出所提模型与其他方法的对比实验结果，并对实验结果进行相应的分析。

3.1 数据集与度量标准

MovieLens[1]：明尼苏达计算机科学系 GroupLens 搜集的电影评分数据集。本实验选取 138493 个用户在 1995-1-9 到 2015-3-31 期间对 27278 部电影做出的 20000263 条评分记录(包含时间戳记录)，稀疏度为 0.540%。通过将用户消费物品数量作为过滤条件对数据进行统计可以得到：用户消费物品数量低于 15 的数据稀疏度为 0，另外用户消费数量不大于 50、不大于 100 以及大于或等于 100 的数据稀疏度分别为 0.257%，0.320%和 1.144%。考虑到冷启动的问题，后面的实验针对用户消费历史数据较少的用户进行了对比分析，发现所提模型与其他模型相比表现更好。

豆瓣图书评分数据集[2]：北京大学信息科学与技术学院发布于北京大学开放数据平台的图书评分数据集。本实验选取 383033 个用户对 89908 本书籍的 13506215 条评分记录(不包含时间戳记录)，稀疏度为 0.012%。同样通过将用户消费物品数量作为过滤条件进行数据统计可以得到：用户消费数量不大于 5、不大于 10 以及大于或等于 10 的数据稀疏度分别为0.005%，0.006%和 0.034%。

基于相近的优化目标[10]，使用相同的离线度量指标来比较不同的模型：召回率 Recall@M、归一化折扣累计增益 NDCG@M 以及平均准确率均值 MAP@M。对于每一个用户，所有的指标都是将未观测的物品预测值与它们的真实值相比较。另外，上述指标都是对测试集中的所有用户求均值。

形式上定义 π 来表示所有物品，如果用户消费了物品 $\pi(i)$，则 $u(\pi(i))$等于 1。

准确率(Precision)和召回率(Recall)是推荐系统中常用

1) http://grouplens.org/datasets/movielens/20m/

2) http://opendata.pku.edu.cn/dataset.xhtml?persistentId=doi:10.18170/DVN/LA9GRH

的分类指标。如表 1 所列,可以得出单个用户的准确率和召回率的形式化定义分别为 $P_u=TP/(TP+FP)$ 和 $R_u=TP/(TP+FN)$。在实证研究中使用式(13)表示召回率,分子表示在物品排序列表前 M 中用户实际消费过的物品;对于分母表达式,选择用户推荐列表长度 M 和用户 u 消费物品数目之间的最小值。召回率 Recall@M 最大值 1 对应表示长度是 M 的推荐列表上的物品都与用户偏好相关。另外,正确率通常只考虑返回结果中相关物品的个数,没有考虑物品之间的排序。考虑到结果是有序列表,需要求得结果中每个位置的 Precision,之后对所有位置的 Precision 求均值,即是公式AP@M(15)的定义。MAP@M 是计算所有用户平均精度(AP)的均值。

表 1　需要预测物品的 4 种情况

是否喜欢	系统推荐	系统未推荐
喜欢	TPFN	FPTN
不喜欢	FPTN	TPFN

DCG 的主要思想是用户喜欢的物品被排在前面比排在后面能更大程度上提高用户体验。式(14)的分子可以理解为推荐列表中每一个物品的增益,该值随用户心理预期的排位递减(这里是指数递减),分母可以理解为排名不同的物品的增益应该赋予不同的折算因子,排名越靠后则对应的折算越多,即最后的增益越小。

用户 u 的 Recall@M 定义为:

$$\text{Recall@}M(u,\pi):=\sum_{i=1}^{M}\frac{I\{u(\pi(i))=1\}}{\min(M,\sum_{i'}^{I}I\{u(\pi(i'))=1\}} \tag{13}$$

用户 u 的 DCG@M 定义为:

$$\text{DCG@}M(u,\pi):=\sum_{i=1}^{M}\frac{2^{I\{u(\pi(i))=1\}}-1}{\log(i+1)} \tag{14}$$

用户 u 的 AP@M 定义为:

$$\text{AP@}M(u,\pi):=\sum_{i=1}^{M}\frac{\text{Precision@}i(u,\pi)}{\min(i,\sum_{i'}^{I}I\{u(\pi(i'))=1\}} \tag{15}$$

3.2　实验设计和结果分析

通过 2.2 节描述的推理算法对所提出的模型进行训练。对于不同的模型,将根据验证集而计算得出的度量指标 NDCG@100 作为算法收敛的判断标准。

实验通过分别比较加权矩阵分解模型(WMF)[10]、WMF+Amend bias(WMFAB)用户偏见修正的加权矩阵分解模型、WMF+ itemsbedding(Cofactor)[10] 联合分解模型和模型 WMF+itemsbedding+Amendbias(Our Model)来表明所提模型的有效性,我们强调不同的模型充分利用相同的数据并优化相似的目标。Our Model 受益于用户偏见修正项和物品共现正则项。

鉴于 MovieLens 数据集包含时间戳,首先对数据按时间先后进行排序,然后按 70%,10%,20%的比例得到训练集、验证集和测试集;训练集和测试集的时间段分别是 1996-01-29 08 时到 2009-10-28 11 时和 2009-10-28 13 时到 2015-03-31 02 时。对于没有时间戳的豆瓣图书评分数据集,随机按 70%,10%和 20%的比例选择训练集、验证集以及测试集。

首先主要选取基于排序的指标 NDCG 来分析不同隐式特征空间维度下模型的变化。对于MovieLens数据集来说(见图 1),在不同特征维度下,WMF 模型和 WMFAB 模型都出现总体下降的趋势,Cofactor 模型先在维度小于 100 内有提高,维度大于 100 之后反而下降。而 Our Mo-del在维度小于 100 内有明显提高,维度大于 100 后基本保持不变。另外,本文提出的用户偏见修正的加权矩阵分解模型 WMFAB 在维度为 25 时优于其他模型,这说明修正用户偏见对于加权矩阵分解模型 WMF 可以起到改进的作用。另外,对于豆瓣数据集来说(见图 2),WMF,WMFAB,Cofactor 以及 OurModel 在不同特征维度上呈整体上升的趋势;WMFAB 在维度 50 以内时,明显优于 WMF 和 Cofactor 模型,而当维度大于 50 以后,Cofactor 模型优于 WMF 以及 WMFAB 模型。OurModel 在不同特征维度上总体优于其他模型。

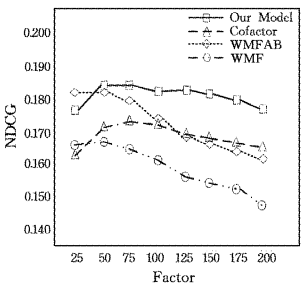

图 1　MovieLens_20M

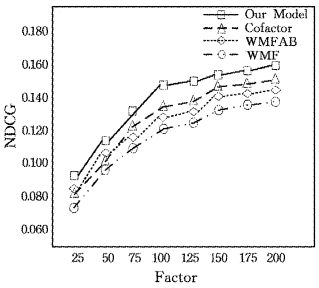

图 2　Douban_books

借鉴文献[10,15]将不同模型的隐式特征空间维度统一设置为 100,并在上述基础上得出 MovieLens 电影评分数据集和豆瓣图书评分数据集在不同度量指标上的实验结果数据(见表 2 和表 3),由表可知本文模型在不同的指标上都得到了提高。

表 2　在 MovieLens_20M 数据集上的结果

	Recall@20	Recall@50	NDCG@100	MAP@100
WMF	0.1330	0.1650	0.1600	0.0470
WMF+amendbias	0.1437	0.1690	0.1649	0.0500
WMF+itembedding	0.1448	0.1765	0.1724	0.0545
OurModel	**0.1673**	**0.1836**	**0.1820**	**0.0619**

表 3　在 Douban_books 数据集上的结果

	Recall@20	Recall@50	NDCG@100	MAP@100
WMF	0.1395	0.1996	0.1270	0.0650
WMF+amendbias	0.1421	0.2007	0.1276	0.0671
WMF+itembedding	0.1565	0.2149	0.1383	0.0734
OurModel	**0.1631**	**0.2201**	**0.1437**	**0.0793**

考虑到不同的模型都需要参数调优,首先根据 WMF 模型在验证集上的表现选取超参数 c_{ui} 和正则化参数$\lambda_\theta=\lambda_\beta$ 为定值。用户偏见修正的加权矩阵分解模型保持上述参数不变,至于联合分解模型,同样根据验证集上的效果选择比例系数 l,图 3 和图 4 给出了本文模型在不同数量级下的性能表现。由图可知,本文联合分解模型在系数数量级较小时的性能明显优于数量级变大的情况。

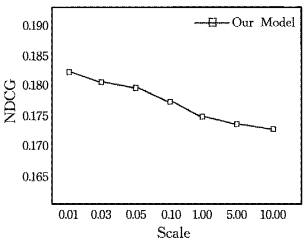

图 3　MovieLens_20M

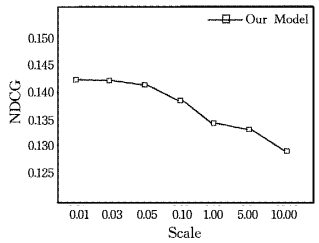

图 4　Douban_books

另外,为了得到更加细粒度的分析,变化训练集和测试集的不同分割比例(20%到 90%)来比较本文模型和其他模型的性能差异,特征维度和比例参数保持不变,结果如图 5 和图

6 所示。从图中可知，Cofactor 模型总体上优于 WMF 模型和 WMFAB 模型，Our Model 总体上优于 WMF，WMFAB 和 Cofactor 模型；训练样本取 20%时表现最差，在 MovieLens 数据集中与 Cofactor 模型基本等价，而在豆瓣数据集中与 WMFAB 基本等价。另外，训练样本分别在 MovieLens 数据集中取 40%时以及在豆瓣数据集中取 80%时性能有显著的提高，前者比 WMF 模型和 Cofactor 模型分别提高 4.93% 和 4.19%，后者比 WMF 模型和 WMFAB 模型分别提高 1.67% 和 1.61%。

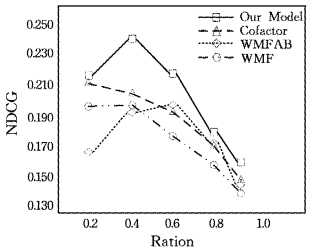

图 5 MovieLens_20M

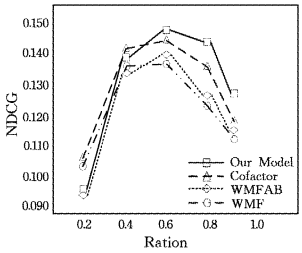

图 6 Douban_books

最后，考虑到冷启动问题一直是协同过滤中的经典问题，以较低的用户活跃度（用户消费物品数目较少）来分析物品冷启动问题。对于 MovieLens 数据集来说，选择将物品消费数量 100 以内的用户二分为“1～50”和“50～100”来分析模型的表现。如图 7 所示，当用户消费物品数量在“1～50”区间（黑色条形）时，Cofactor 模型的效果要优于 WMFAB 模型和 WMF 模型；而当消费数量在“50～100”区间（白色条形）时，WMFAB 模型的效果优于 Cofactor 模型和 WMF 模型。然而与 WMF，WMFAB 和 Cofactor 模型相比，Our Model 在用户消费物品数量较少的情况下，均优于其他 3 种模型。对于豆瓣数据集来说，我们则选择将消费数量 10 以内的用户分为“1～5”和“5～10”，如图 8 所示，在用户消费数量在“1～5”区间（黑色条形）和“5～10”区间（白色条形）时，WMFAB 模型优于 Cofactor 模型和 WMF 模型，这进一步说明了修正用户偏见对于传统模型的改进作用，而 Our Model 在上述不同区间又明显优于其他 3 种模型。上述分析说明，本文模型在冷启动问题上有更好的表现。

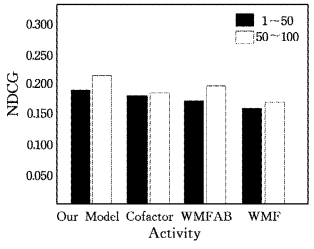

图 7 MovieLens_20M

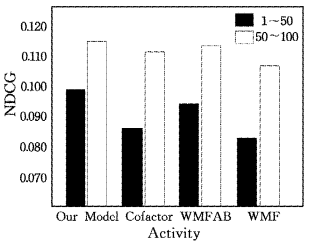

图 8 Douban_books

结束语 本文提出了一种新的模型，即在矩阵分解基础上分别加入用户偏见修正和物品共现信息来学习用户因子和物品因子。实验证明了这种修正用户偏好的联合分解形式在不同的推荐指标上带来了性能提升。在以后的工作中，我们还将考虑将该模型与其他模型相结合，从而提出更高性能的混合模型。

参 考 文 献

[1] SINGH A P, GORDON G J. Relational Learning via Collective Matrix Factorization[J]. ACM Sigkdd International Conference on Knowledge Discovery & Data Mining, 2008, 40(46): 650-658.

[2] RISH I, GRABARNIK G, CECCHI G, et al. Closed-Form Supervised[J/OL]. http://www.mendeley.com/research-papers/closedform-supervised-dimensionality-reduction-generalized-linear-models.

[3] MA H, YANG H, LYU M R, et al. SoRec: Social Recommendation Using Probabilistic Matrix Factorization[C]// ACM Conference on Information & Knowledge Management. 2008: 931-940.

[4] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed Representations of Words and Phrases and their Compositionality [J]. Advances inNeural Information Processing Systems, 2013, 26: 3111-3119.

[5] LEVY O, GOLDBERG Y. Neural word embedding as implicit matrix factorization[J]. Advances in Neural Information Processing Systems, 2014, 3: 2177-2185.

[6] KOREN Y, BELL R, VOLINKSKY C. Matrix Factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.

[7] SHI J P, WANG N Y, XIA Y, et al. SCMF: Sparse Covariance Matrix Factorization for Collaborative Filtering[C]// IJCAI. 2013.

[8] KOREN Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model[C]// ACM Sigkdd International Conference on Knowledge Discovery & Data Mining. 2008: 426-434.

[9] SINGH A P, GORDON G J. A Unified View of Matrix Factorization Models[M]// Machine Learning & Knowledge Discovery in Databases. Spinger-Verlag Berlin Heidelberg, 2008, 5212: 358-373.

[10] LIANG D, ALTOSAAR J, CHARLIN L, et al. Factorization Meets the Item Embedding: Regularizing Matrix Factorization with item Co-occurrence[C]// ACM Conference on Recommender System. 2016: 59-66.

[11] BARKAN O, KOENIGSTEIN N. Item2Vec: Neural Item Embedding for Collaborative Filtering[OL]. http://arXiv.org/obs/1603.04259.

[12] LIANG D, ZHAN M, ELLIS D P W. Content-aware collaborative music recommendation using pre-trained neural networks [C]// Proceedings of the 16th International Society for Music Information Retrieval Conference. 2015: 295-301.

[13] ALMAHAIRI A, KASTNER K, CHO K, et al. Learning distributed representations from reviews for collaborative filtering[C]// ACM Conference. 2015: 147-154.

[14] HE R N, MCAULEY J J. Visual Bayesian personalized ranking from implicit feedback[OL]. http://cseweb.ucsd.edu/~jmcauley/pdfs/aaai16.pdf.

[15] HU Y, KOREN Y, VOLINSKY C. Collaborative Filtering for Implicit Feedback Datasets[C]// Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM'08). 2008: 263-272.

[16] BOUCHARD G, YIN D, GUO S. Convex collective matrix factorization[C]// AISTATS. 2013.

[17] 崔斌. 豆瓣图书评分数据[OL]. http://dx.doi.org/10.18170/DVN/LA9GRH.