

基于成对交互张量分解的标签推荐

鲁亚男, 杜东舫

(中国科学技术大学计算机科学技术学院, 安徽合肥 230027)

摘要: 标签推荐系统是为目标用户推荐最可能用来标记某个资源的一系列标签。目前基于塔克分解模型, 相比传统的 FolkRank 等算法具有更好的预测质量, 但它本身的时间复杂度很高, 很难适用于大中型数据集; 而正则分解模型的时间复杂度虽然为线性, 但预测质量并不高。针对上述问题, 在改进塔克分解模型的基础上首先提出成对交互张量分解模型 PITD。该模型仅考虑用户、资源和标签 3 个特征之间的部分两两交互关系, 减少了无关信息对模型性能以及效率的影响。进而, 利用贝叶斯个性化排序方法对 PITD 模型进行推导, 并设计了相应的优化算法。最后, 在真实数据集上的广泛实验表明, PITD 模型对比算法具有更好的推荐性能。

关键词: 推荐系统; 标签推荐; 张量分解; BPR

中图分类号: TP391 **文献标识码:** A doi: 10.3969/j.issn.0253-2778.2019.01.005

引用格式: 鲁亚男, 杜东舫. 基于成对交互张量分解的标签推荐[J]. 中国科学技术大学学报, 2019, 49(1): 31-39.
LU Yanan, DU Dongfang. Pairwise interaction tensor factorization based tag recommendation[J].
Journal of University of Science and Technology of China, 2019, 49(1): 31-39.

Pairwise interaction tensor factorization based tag recommendation

LU Yanan, DU Dongfang

(School of Computer Science and Technology, USTC, Hefei 230027, China)

Abstract: The tag recommendation system is a series of tags that are most likely to be used to tag a resource for the target user. Currently, the Tucker decomposition model has better prediction quality than the traditional FolkRank algorithm, but it has high time complexity and is difficult to apply to large and medium-sized data sets. Although the time complexity of the regular decomposition model is linear, its prediction quality is not high. To solve these problems, firstly, the paired interaction tensor decomposition model PITD on the basis of improving the Tucker decomposition model is proposed. The model considers only some of the two-to-two interactions between the three characteristics of users, resources, and tags, reducing the impact of irrelevant information on model performance and efficiency. Then, the PITD model is deduced by Bayesian personalization method, and the corresponding optimization algorithm is designed. Finally, extensive experiments on real data sets show that the PITD model has better recommendation performance than the comparison algorithm.

Key words: recommendation system; tag recommendation; tensor decomposition; BPR

0 引言

随着互联网的迅速发展,涌现了海量的信息数据.为了使用户方便获取有价值信息,推荐系统应运而生.推荐系统通过对用户资源交互矩阵等信息进行处理分析,得到相关的个人偏好特征,进而向用户进行个性化推荐服务.应用最广泛的推荐方法是协同过滤算法^[1-3],其基本思想是假设相似偏好的用户所喜欢的资源也应该是相似的.协同过滤算法大致分为如下两类:基于内存的协同过滤算法^[1]和基于模型的协同过滤算法^[4-5].基于内存的协同过滤算法通过用户评分向量等信息寻找相似用户,并依据相似用户对相关资源的评分预测待推荐用户对未评分资源的评分.基于模型的协同过滤算法则通过分析用户行为记录等信息建立偏好模型,从而训练模型得到用户及资源的偏好特征,并通过计算用户和资源特征之间的相似程度进行个性化推荐.

传统的推荐系统通常考虑用户和资源之间的二维交互关系,如用户的点击、评分等.这种方法将模型预测得分较高的前 N 个资源,作为最终结果推荐给目标用户.现实世界中,除了用户资源间的二维关系,很多推荐场景往往需要综合考虑更高维度的交互行为.相比于二维推荐系统,高维推荐适用于更多的现实场景,同时也能够更好地融合多种信息.

近年来不断涌现的 Delicious、last.fm 以及其他新兴社会标记系统使得用户可以自由地对网络资源(如图书)添加标签.而标签推荐系统^[6-7]作为社会标注系统中有迫切需求且十分重要的关键技术,越来越多地受到研究人员的关注^[8-15].不同于之前提到的二维推荐系统,这种场景下的标签推荐系统需要考虑用户、资源以及标签之间的三维交互关系.数据的高维特性以及数据间的关系复杂性为标签推荐算法的设计增加了难度.标签推荐中的关键问题是需要找到合适的结构描述用户、资源以及标签间的三维关系,并采用有效的算法对该三维关系进行分解,同时在降维过程中要尽量避免用户、资源和标签三维数据间整体交互信息的丢失.

为了解决上述问题,本文提出了一种基于成对交互张量分解的标签推荐模型(pairwise interaction tensor factorization, PITD).该模型在将用户、资源以及标签之间的三维数据存储在张量结构的基础上,仅考虑用户、资源和标签三个特征之间的部分两两交互关系,减少了无关信息对模型性能以及效率的影响.进而,利用贝叶斯个性化排序^[16](Bayesian

personalized ranking, BPR)算法对 PITD 模型进行推导,并设计了相应的优化算法.本文的主要贡献在于:

(I)针对标签推荐系统的特点提出了一种有效刻画其特性的基于成对交互张量分解的标签推荐模型 PITD.

(II)提出了基于贝叶斯个性化排序(BPR)的 PITD 标签推荐算法.通过推导 PITD 模型,并对参数进行优化求解,得到了最终的推荐方案.

在真实数据集上的广泛实验验证了 PITD 标签推荐算法在不同特征维度以及不同推荐指标下均能够显著提升标签推荐性能.

1 相关工作

考虑到社会标签系统具有相互关联的用户、资源和标签 3 个对象,文献[17]将 3 部图结构引入了标签系统中.该文献将用户、资源和标签作为图中的节点,并根据 3 元组<用户,资源,标签>的两两共现关系向图中添加带权边.其大小取决于用户对资源使用的标签个数.例如,如果用户 u 对资源 i 使用了两个标签 t_1 和 t_2 ,那么图中的两条边 (u, i, t_1) 和 (u, i, t_2) 的权重都将被赋值为 $1/2$.进而,该文献将 3 部图转化为二部图,并在此基础上进行处理以解决最终的标签推荐问题.这种传统的转换方式往往会丢失 3 个对象间原本存在的整体关系,从而在一定程度上限制标签推荐性能.

文献[18]首先提出了 PageRank 算法,该算法通过计算网络中各个页面的 PageRank 值,并将该值的大小作为页面重要性的评判标准,来对网络中的各个页面进行排序.由于 PageRank 并不具备个性化的特点,直接用于标签推荐也不能取得较高的推荐质量.

文献[19]提出了 FolkRank 算法,该算法的基本思想是:若一个资源被重要的用户标注过,并且该用户在标注时所使用的标签也是重要的,则该资源将是重要的.此外,对于用户和标签,该规则同样成立.与 PageRank 类似, FolkRank 也采用了随机冲浪模型,且由于提高了自身的个性化程度,从而具有更好的推荐质量.由于该算法对每个<用户,资源>对都要重新计算标签推荐列表,因此它的运行时间复杂度过高,从而当网络中的信息量呈爆炸式增长时,其适用性将不断下降.

上述标签推荐算法均将三维空间中的用户、资源以及标签间的 3 元关系划分成二维空间中的 3 组

双边关系,即用户与资源、用户与标签以及资源与标签.虽然通过该转化可以将已有的二维推荐算法应用于标签推荐系统,但却因此丢失了用户、资源和标签之间的整体交互信息.针对此问题,文献[20]首先将张量结构引入了社会性标签系统中,并用张量的3个维度分别表示社会性标签系统中的用户、资源和标签3个对象.进而将二维矩阵分解推广到三维,得到了HOSVD(高阶奇异值分解)方法,最终通过对原张量中缺失值的预测,重构原张量,并据此实现标签推荐.由于真实数据集的高度稀疏性,该算法并不能达到很高的精度.

文献[6]提出了一种更优的基于塔克分解(Tucker decomposition, TD)的标签推荐方法.同时,由于文献[20]中的HOSVD方法存在数据翻译简单且缺少正则化项的缺陷,该文献定义了带正则化项的新目标函数,同时对原始数据的翻译方法也做了改进,进而提出了优化求解TD模型的RTF(ranking with tensor factorization)方法.实验结果表明RTF不仅比HOSVD更高效,同时推荐效果也更好.

正则分解(canonical decomposition, CD)是TD的一种特殊情况^[21],也是一种经典的张量分解模型.它通过将TD模型的核心张量变为一个特殊的超对角张量来简化TD模型.虽然该模型具有线性的时间复杂度,但它对三维数据分解的解并非必然存在,这使得CD模型并没有在实际中广泛应用.

虽然基于TD模型的张量分解模型与其他方法相比,已经展现出了较高的标签推荐质量.由于该模型分解得到的核心张量仍是三维的,这就导致模型训练和预测的时间复杂度过高,进而使得TD模型不能很好地应用于大中型数据集.针对该问题,文献[22]提出了PITF(pairwise interaction tensor factorization)分解模型.该模型是CD模型的一种特殊情况,它以不考虑用户和资源之间的交互来简化分解,最终得到线性模型,从而大大减小了算法的复杂度.同时,相关实验表明,PITF模型比传统的CD模型具有更好的推荐质量.

2 基于成对交互张量分解的标签推荐模型

2.1 问题定义及符号说明

为了对个性化标签推荐进行形式化表示,我们采用文献[6]中的表示方法,即用 U 表示系统中所

有用户, I 表示系统中所有资源, T 表示系统中所有标签. S 是历史标记信息,即 $S \subseteq U \times I \times T$, $(u, i, t) \in S$ 就意味着用户 u 曾经用标签 t 标记过资源 i .本文的目的是为每个用户资源对 $\langle u, i \rangle$ 找到一个针对所有标签的个性化全序关系 $\succ_{u,i}$,其满足偏序关系的3条基本性质:

(I)完全性

$$\forall t_1, t_2 \in T: t_1 \neq t_2 \Rightarrow t_1 \succ_{u,i} t_2 \vee t_2 \succ_{u,i} t_1;$$

(II)反对称性

$$\forall t_1, t_2 \in T: t_1 \succ_{u,i} t_2 \wedge t_2 \succ_{u,i} t_1 \Rightarrow t_1 = t_2;$$

(III)传递性

$$\forall t_1, t_2, t_3 \in T: t_1 \succ_{u,i} t_2 \wedge t_2 \succ_{u,i} t_3 \Rightarrow t_1 \succ_{u,i} t_3.$$

具体地,定义4元组 (u, i, t_A, t_B) ,表示用户在给资源 i 贴标签时选择了标签 t_A ,而没有选择标签 t_B .由此,可以定义如下的训练集:

$$D_S = \{(u, i, t_A, t_B) \mid (u, i, t_A) \in S \wedge (u, i, t_B) \notin S\}.$$

在给定训练集的情况下,本文的目的是预测用户 u 在对某个指定资源 i 进行标记时使用标签 t 的可能性(即对该标签的偏好程度) $\hat{y}_{u,i,t}$,进而根据该评分得到可以推荐给用户 u 的关于某个指定资源 i 的TopN标签列表:

$$\text{Top}(u, i, N) = \underset{t \in T}{\text{argmax}} \hat{y}_{u,i,t}.$$

其中, N 表示最终推荐给 \langle 用户,资源 \rangle 对 (u, i) 的标签数.

2.2 PITD模型介绍

PITD是对TD模型的改进,TD模型的具体形式为

$$Y = C \times_U U \times_I I \times_T T \quad (1)$$

式中, \times_U 是张量与矩阵进行乘法运算的运算符,下标 U 表示张量与矩阵相乘时张量的展开方向,即 $Y = C \times_U U$ 表示

$$Y_{u,i,t} = \sum_{u'=1}^{k_u} C_{u',i,t} \cdot U_{u,u'} \quad (2)$$

具体的改进方案是:不再考虑TD模型中用户和资源间的任何交互.因为本文在设计模型以及最后对结果进行预测排序时,都没有考虑过这二者之间的任何交互关系,而是将它们看作一个整体(记为post)进行推荐.在不考虑用户和资源间的交互后,分解模型得到的核心张量及3个特征矩阵之间的关系也会发生变化.具体地,我们得到的模型为

$$\hat{Y} = C \times_U [U \oplus 1] \times_I [1 \oplus I] \times_T [T^U \oplus T^I] \quad (3)$$

式中，“ \oplus ”是用来拼接矩阵的操作. 为了进一步保证用户和资源之间没有任何交互, 进而完成最终的预测, 我们也需要对核心张量的结构做一些改变. 如图 1 所示, 核心张量 C 被分解成八个部分, 其中仅主对角线的两个部分 C^1 和 C^2 最重要, 其余次要部分均为 0; 而图中其余 3 个部分依次对应 $[U \oplus 1]$ 、 $[1 \oplus I]$ 以及 $[T^U \oplus T^I]$. 将图 1 所示的张量分解方法展开后得到

$$\hat{Y} = C^1 \times_U U \times_I 1 \times_T T^U + C^2 \times_U 1 \times_I I \times_T T^I \quad (4)$$

其标量表示形式为

$$\hat{y}_{u,i,t} = \sum_{u'} \sum_{i'} \sum_{t'} c_{u',i',t'}^1 \cdot u_{u,u'} \cdot 1_{i,i'} \cdot t_{t,t'}^U + \sum_{u'} \sum_{i'} \sum_{t'} c_{u',i',t'}^2 \cdot 1_{u,u'} \cdot i_{i,i'} \cdot t_{t,t'}^I \quad (5)$$

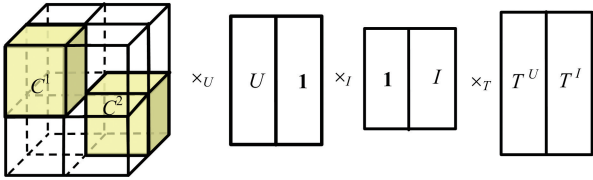


图 1 PITD 模型

Fig. 1 PITD model

通过利用基于 BPR 的标签推荐算法可以求得最佳参数 C^1 、 C^2 、 U 、 I 、 T^U 和 T^I , 从而完成对标签的预测.

2.3 基于 BPR 的 PITD 标签推荐算法

我们在上节已经将为用户推荐一系列排序好的、用户最可能用到的标签任务转化为对某个给定的 $\text{post}(u, i)$ 寻找最佳标签排序 $\succ_{u,i} \subset T \times T$ 的任务, 从而需要优化模型参数, 以使下式的概率最大.

$$p(\theta | \succ_{u,i}) \propto p(\succ_{u,i} | \theta) \cdot p(\theta) \quad (6)$$

式中, θ 是模型参数. 假设 post 之间是相互独立的, 则通过对模型参数使用最大后验估计可得

$$\arg\max_{\theta} \prod_{(u,i) \in U \times I} p(\succ_{u,i} | \theta) \cdot p(\theta) \quad (7)$$

接下来, 我们首先对 $p(\succ_{u,i} | \theta)$ 进行分析. 假设当 $t_A \neq t_C$ 且 $t_B \neq t_D$ 时, $p(t_A \succ_{u,i,t_B} | \theta)$ 与 $p(t_C \succ_{u,i,t_D} | \theta)$ 之间相互独立. 由于 $t_A \succ_{u,i,t_B}$ 是一个伯努利实验, 因此有

$$\prod_{(u,i) \in U \times I} p(\succ_{u,i} | \theta) = \prod_{(u,i,t_A,t_B) \in U \times I \times T^2} p(t_A \succ_{u,i,t_B} | \theta) \delta((u,i,t_A,t_B) \in D_S) \cdot (1 - p(t_A \succ_{u,i,t_B} | \theta)) \delta((u,i,t_B,t_A) \in D_S) \quad (8)$$

式中, δ 是指示函数, 用来表达 4 元组 (u, i, t_A, t_B) 是否在训练集中, 我们将其定义为

$$\delta(b) = \begin{cases} 1, & \text{如果 } b \text{ 成立;} \\ 0, & \text{否则.} \end{cases}$$

由于我们所预测的评分函数满足全序关系, 因此 $(u, i, t_B, t_A) \in D_S$ 是不可能的, 从而式(8)中的第二项是不可能出现的, 于是式(8)就可以被简化为

$$\prod_{(u,i) \in U \times I} p(\succ_{u,i} | \theta) = \prod_{(u,i,t_A,t_B) \in D_S} p(t_A \succ_{u,i,t_B} | \theta) \quad (9)$$

为了估计模型参数 θ , 定义

$$p(t_A \succ_{u,i,t_B} | \theta) = \sigma(\hat{y}_{u,i,t_A,t_B}(\theta)) \quad (10)$$

式中, σ 是 logistic 函数: $\sigma(x) = 1/(1 + e^{-x})$. $\hat{y}_{u,i,t_A,t_B}(\theta)$ 代表用户 u 在看了资源 i 之后, 面对标签 t_A 和标签 t_B 时, 选择标签 t_A 而非标签 t_B 的偏好得分.

假设这个偏好得分是用户资源组 $\langle u, i \rangle$ 对标签 t_A 的偏好值减去用户资源组 $\langle u, i \rangle$ 对标签 t_B 的偏好值, 并借助 2.2 节提出的 PITD 模型, 我们定义

$$\hat{y}_{u,i,t_A,t_B}(\theta) = \hat{y}_{u,i,t_A}(\theta) - \hat{y}_{u,i,t_B}(\theta) \quad (11)$$

式中, θ 即为 C^1 、 C^2 、 U 、 I 、 T^U 和 T^I . 为了方便表示, 我们在接下来的部分省略 θ . 从而得到

$$\prod_{(u,i) \in U \times I} p(\succ_{u,i} | \theta) = \prod_{(u,i,t_A,t_B) \in D_S} \sigma(\hat{y}_{u,i,t_A} - \hat{y}_{u,i,t_B}) \quad (12)$$

为了求得后验概率, 还需要得到先验概率 $p(\theta)$. 假设模型参数 θ 服从期望为 0, 方差为 σ_{θ}^2 的正态分布, 即 $\theta \sim N(0, \sigma_{\theta}^2 I)$. 对于均值为 0, 方差为 $\sigma_{\theta}^2 I$ 的 D 维高斯分布而言, 其概率密度函数为

$$f(x) = (2\pi)^{-\frac{D}{2}} \cdot (\sigma_{\theta}^2)^{-\frac{D}{2}} \cdot e^{-\frac{1}{2}[(\sigma_{\theta}^2)^{-1} \cdot x' \cdot x]} \quad (13)$$

于是

$$\begin{aligned} \ln p(\theta | \sigma_{\theta}^2) &= \ln \left[\prod_{i=1}^N N(\theta_i | 0, \sigma_{\theta}^2 I) \right] = \\ &= \sum_{i=1}^N \ln \left[(\sigma_{\theta}^2)^{-\frac{D}{2}} \cdot e^{-\frac{1}{2}[(\sigma_{\theta}^2)^{-1} \cdot \theta_i' \cdot \theta_i]} \right] + C_1 = \\ &= -\frac{1}{2\sigma_{\theta}^2} \sum_{i=1}^N \theta_i' \theta_i - \frac{1}{2} N D \ln \sigma_{\theta}^2 + C_1 \quad (14) \end{aligned}$$

式中, C_1 为常数.

将上面得到的式(12)和(14)带入式(6), 就得到了基于 BPR 的 PITD 标签推荐算法的目标函数.

$$\begin{aligned}
\text{BPR-OPT:} &= \ln \prod_{(u,i,t_A,t_B) \in D_S} \sigma(\widehat{y}_{u,i,t_A,t_B}) \cdot p(\theta) = \\
& \sum_{(u,i,t_A,t_B) \in D_S} \ln \sigma(\widehat{y}_{u,i,t_A,t_B}) + \ln p(\theta) = \\
& \sum_{(u,i,t_A,t_B) \in D_S} \ln \sigma(\widehat{y}_{u,i,t_A,t_B}) - \frac{1}{2\sigma_\theta^2} \sum_{i=1}^N \theta_i' \theta_i - C_2 = \\
& \sum_{(u,i,t_A,t_B) \in D_S} \ln \sigma(\widehat{y}_{u,i,t_A,t_B} - \widehat{y}_{u,i,t_B}) - \lambda_\theta \|\theta\|_F^2
\end{aligned} \tag{15}$$

式中, $\lambda_\theta = 1/2\sigma_\theta^2$, $\|\theta\|_F^2 = \sum_{i=1}^N \theta_i' \theta_i$, C_2 是常数, 对结果无影响, 所以最终表达式将其忽略. 通过最大化后验概率, 就可以求得参数 θ 即 C^1, C^2, U, I, T^U 和 T^I .

2.4 学习算法

为了最大化后验概率, 可以采用梯度上升方法, 具体的, 容易得到

$$\begin{aligned}
\frac{\partial}{\partial \theta} (\ln \sigma(\widehat{y}_{u,i,t_A,t_B}) - \lambda_\theta \cdot \|\theta\|_F^2) &\propto \\
(1 - \sigma(\widehat{y}_{u,i,t_A,t_B})) \cdot \\
\left(\frac{\partial \widehat{y}_{u,i,t_A}}{\partial \theta} - \frac{\partial \widehat{y}_{u,i,t_B}}{\partial \theta} \right) - \lambda_\theta \cdot \theta
\end{aligned} \tag{16}$$

将上式对 PITD 模型中的各个参数求偏导数, 可得

$$\begin{aligned}
\frac{\partial \widehat{y}_{u,i,t}}{\partial c_{u',i',t'}^1} &= u_{u,u'} \cdot t_{i,i'}^U, \quad \frac{\partial \widehat{y}_{u,i,t}}{\partial c_{u',i',t'}^2} = i_{i,i'} \cdot t_{i,i'}^I, \\
\frac{\partial \widehat{y}_{u,i,t}}{\partial u_{u,u'}} &= \sum_{i'} \sum_{t'} c_{u',i',t'}^1 \cdot t_{i,i'}^U, \\
\frac{\partial \widehat{y}_{u,i,t}}{\partial i_{i,i'}} &= \sum_{u'} \sum_{t'} c_{u',i',t'}^2 \cdot t_{i,i'}^I, \\
\frac{\partial \widehat{y}_{u,i,t}}{\partial t_{i,i'}^U} &= \sum_{u'} \sum_{t'} c_{u',i',t'}^1 \cdot u_{u,u'}, \\
\frac{\partial \widehat{y}_{u,i,t}}{\partial t_{i,i'}^I} &= \sum_{u'} \sum_{t'} c_{u',i',t'}^2 \cdot i_{i,i'}.
\end{aligned}$$

由于 D_S 大小为 $O(|S| \cdot |T|)$, 通常非常巨大, 因此直接采用梯度上升算法将会十分耗时. 按照用户或者 post 的顺序直接使用随机梯度上升算法也不可行. 为了解决这个问题, 算法采用均匀采样方法从 D_S 中随机选取四元组, 从而得到最终的学习算法如下:

算法 2.1 基于 BPR 的 PITD 标签推荐算法

输入: D_S, α, λ

输出: C^1, C^2, U, I, T^U 和 T^I

- 1 利用 $N(0, \sigma^2 D)$ 初始化 C^1, C^2, U, I, T^U, T^I
- 2 从 D_S 中均匀采样 (u, i, t_A, t_B)
- 3 $\widehat{y}_{u,i,t_A,t_B} := \widehat{y}_{u,i,t_A} - \widehat{y}_{u,i,t_B}$
- 4 $\delta \leftarrow (1 - \sigma(\widehat{y}_{u,i,t_A,t_B}))$
- 5 for $(u', i', t') \in k_u \times k_i \times k_t$ do
- 6 $C_{u',i',t'}^1 \leftarrow C_{u',i',t'}^1 + \alpha \cdot (\delta \cdot (u_{u,u'} \cdot (t_{t_A,t'}^U - t_{t_B,t'}^U))) - \lambda \cdot C_{u',i',t'}^1$
- 7 $C_{u',i',t'}^2 \leftarrow C_{u',i',t'}^2 + \alpha \cdot (\delta \cdot (i_{i,i'} \cdot (t_{t_A,t'}^I - t_{t_B,t'}^I))) - \lambda \cdot C_{u',i',t'}^2$
- 8 end for
- 9 for $(i', t') \in k_i \times k_t$ do
- 10 $u_{u,u'} \leftarrow u_{u,u'} + \alpha \cdot (\delta \cdot (\sum_{i'} \sum_{t'} c_{u',i',t'}^1 \cdot (t_{t_A,t'}^U - t_{t_B,t'}^U))) - \lambda \cdot u_{u,u'}$
- 11 end for
- 12 for $(u', t') \in k_u \times k_t$ do
- 13 $i_{i,i'} \leftarrow i_{i,i'} + \alpha \cdot (\delta \cdot (\sum_{u'} \sum_{t'} c_{u',i',t'}^2 \cdot (t_{t_A,t'}^I - t_{t_B,t'}^I))) - \lambda \cdot i_{i,i'}$
- 14 end for
- 15 for $(u', i') \in k_u \times k_i$ do
- 16 $t_{t_A,t'}^U \leftarrow t_{t_A,t'}^U + \alpha \cdot (\delta \cdot (\sum_{u'} \sum_{t'} c_{u',i',t'}^1 \cdot u_{u,u'})) - \lambda \cdot t_{t_A,t'}^U$
- 17 $t_{t_B,t'}^U \leftarrow t_{t_B,t'}^U + \alpha \cdot (\delta \cdot (-\sum_{u'} \sum_{t'} c_{u',i',t'}^1 \cdot u_{u,u'})) - \lambda \cdot t_{t_B,t'}^U$
- 18 $t_{t_A,t'}^I \leftarrow t_{t_A,t'}^I + \alpha \cdot (\delta \cdot (\sum_{u'} \sum_{t'} c_{u',i',t'}^2 \cdot i_{i,i'})) - \lambda \cdot t_{t_A,t'}^I$
- 19 $t_{t_B,t'}^I \leftarrow t_{t_B,t'}^I + \alpha \cdot (\delta \cdot (-\sum_{u'} \sum_{t'} c_{u',i',t'}^2 \cdot i_{i,i'})) - \lambda \cdot t_{t_B,t'}^I$
- 20 end for
- 21 重复步骤 2~20, 直到收敛

3 实验结果与分析

3.1 实验数据集

实验中使用了 MovieLens 数据集进行评估. 由于该数据集较大, 实验中将其预处理之后得到 MovieLens 10 记为 M10, MovieLens 19 记为 M19 和 MovieLens 29 记为 M29 这 3 个数据集. 其中, M10 是在完整的 MovieLens 数据集中删除了用户的关联数据(资源的关联数及标签的关联数)少于 10 条的用户(资源及标签). M19 是对完整的 MovieLens 数据集删除了用户的关联数据(资源的关联数及标签的关联数)少于 19 条的用户(资源及标签)得到的. M29 是对完整的 MovieLens 数据集删除了用户的关联数据(资源的关联数及标签的关联数)少于 29 条的用户(资源及标签)得到的. 表 1 对它们的特点进行了总结.

表 1 数据集
Tab. 1 Datasets

数据集	用户数	资源数	标签数	post 数	三元组数
M10 训练集	422	1 281	6 029	14 069	28 147
M10 测试集	419	329	609	419	811
M19 训练集	294	578	4 369	8 799	18 271
M19 测试集	293	211	453	293	582
M29 训练集	202	301	3 107	5 433	11 809
M29 测试集	199	146	374	199	459

3.2 实验设置

对于训练集 S_{train} 和测试集 S_{test} , 本文的划分策略为: 对于每个用户, 随机将它的一个 post 的所有数据从数据集中移出并将其放到 S_{test} 中, 剩下的数据 $S \setminus S_{\text{test}}$ 组成了我们的训练集 S_{train} ; 然后使用 S_{train} 中的数据对模型进行训练学习, 训练完成后, 将训练得到的模型用于 S_{test} , 为 S_{test} 中的每个 post 预测 TopN 列表。

3.2.1 评价指标

实验计算了 Top1 至 Top10 的精确度和召回率, 并最终输出了 F 值作为评判预测质量好坏的标准

$$\text{Prec}(S_{\text{test}}, N) := \frac{\text{avg}_{(u,i) \in P_{S_{\text{test}}}} \frac{|\text{Top}(u,i,N) \cap \{t \mid (u,i,t) \in S_{\text{test}}\}|}{N}}{N}$$

$$\text{Rec}(S_{\text{test}}, N) := \frac{\text{avg}_{(u,i) \in P_{S_{\text{test}}}} \frac{|\text{Top}(u,i,N) \cap \{t \mid (u,i,t) \in S_{\text{test}}\}|}{|\{t \mid (u,i,t) \in S_{\text{test}}\}|}}{N}$$

$$F(S_{\text{test}}, N) := \frac{2\text{Prec}(S_{\text{test}}, N) \cdot \text{Rec}(S_{\text{test}}, N)}{\text{Prec}(S_{\text{test}}, N) + \text{Rec}(S_{\text{test}}, N)}$$

同时, 为了更好地评价标签推荐效果, 本文在 F_1 指标的基础上又增加了常见的 HLU 以及 AUC 指标。

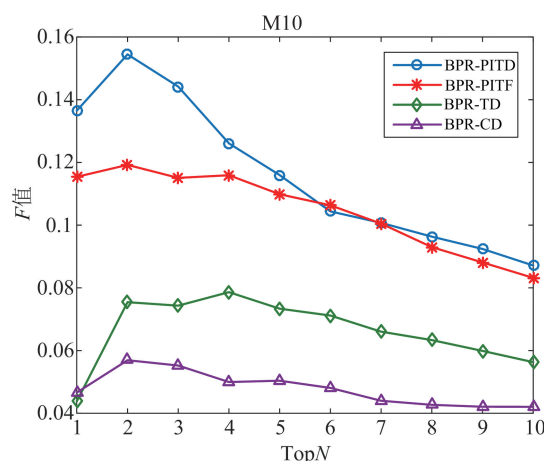
3.2.2 对比算法

鉴于本文算法使用张量方法并结合 BPR 优化过程进行推导, 因此本文的对比算法也选择常见的塔克分解(TD)以及正则分解(CD)并采用 BPR 优化方法进行推导后得到相应的对比算法 BPR-TD 以及 BPR-CD; 此外, 近年来提出的 BPR-PITF 模型在标签推荐方面取得了很好的效果, 本文也将其作为对比算法。

3.3 实验结果

3.3.1 推荐性能评估

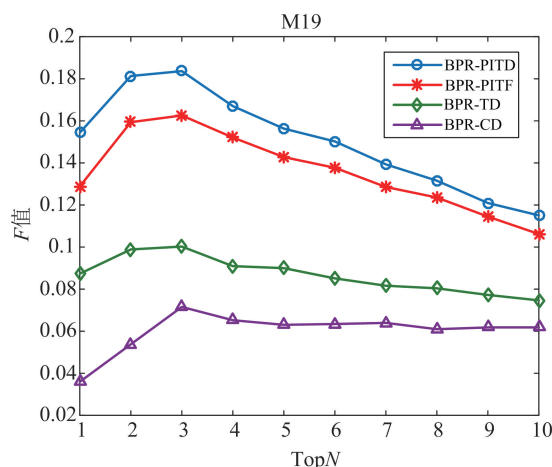
按照上面设计的实验, 对本文 PITD 模型从推荐效果上进行对比验证, 结果如图 2~4 所示。



实验中选取的特征数为 5。

图 2 M10 数据集上 F 值随着 TopN 值变化的变化趋势

Fig. 2 Change trend of F value with TopN on M10 dataset



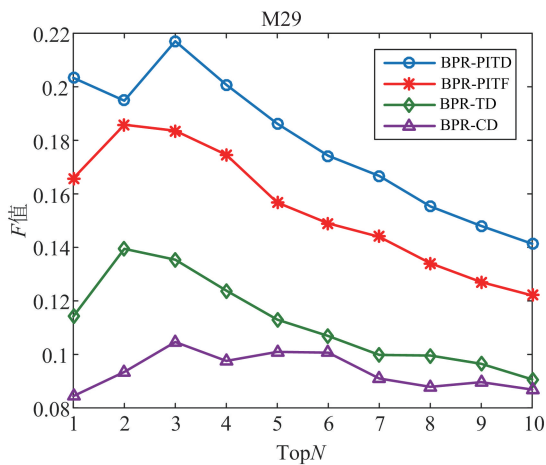
实验中选取的特征数为 5。

图 3 M19 数据集上 F 值随着 TopN 值变化的变化趋势

Fig. 3 Change trend of F value with TopN on M19 dataset

通过观察图 2 至图 4 可以发现, 本文提出的 PITD 模型在不同数据集上都取得了最优的推荐性能. 其中 PITD 优于代表性算法 PITF, 并且显著优于 TD 和 CD 算法. 如推荐 Top1 资源时, 在 M10 数据集上, PITD 对比算法 PITF 提升了 18.3%, 而在 M19 以及 M29 数据集上, 提升效果更是分别达到了 19.8% 和 22.7%. 这充分说明了本文提出算法的正确性与有效性。

此外, 通过观察各个算法的推荐性能与 TopN 值的关系可以发现, F 值几乎都是随着 N 的增大先



实验中选取的特征数为 5.

图 4 M29 数据集上 F 值随着 TopN 值变化的变化趋势

Fig. 4 Change trend of F value with TopN on M29 dataset
增大,后减小,并且在 N 取 2 或者 3 的情况下达到最优.这也反映出特定推荐标签的场景中,系统推荐 2~3 个标签或许比较合适.

表 2,3 分别显示了各模型在不同数据集上的 HLU 以及 AUC 值.从表中可以看出,本文提出的 PITD 模型在各模型集中都取得了最优效果.并且在这两个指标下,PITD 也是优于代表性算法 PITF,同时显著优于 TD 和 CD 算法.这进一步说明了本文提出的 PITD 确实具有更好的推荐性能.

表 2 各模型的 HLU 值

Tab. 2 HLU values of each model

模型	M10 数据集	M19 数据集	M29 数据集
PITF	0.243 671	0.307 613	0.326 181
PITD	0.275 305	0.334 979	0.371 359
TD	0.177 526	0.199 643	0.258 489
CD	0.116 786	0.101 457	0.213 671

表 3 各模型的 AUC 值

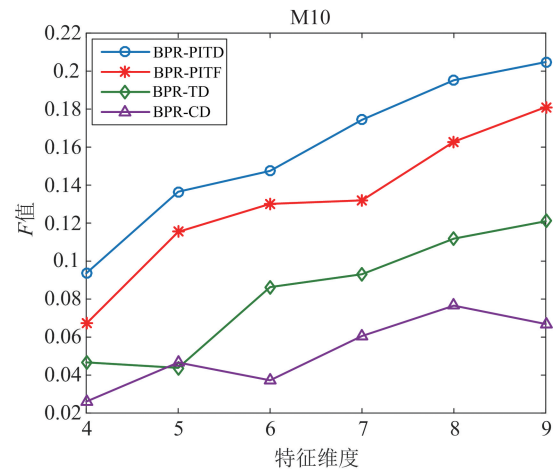
Tab. 3 AUC values of each model

模型	M10 数据集	M19 数据集	M29 数据集
PITF	0.847 496	0.848 277	0.876 997
PITD	0.856 621	0.851 971	0.882 596
TD	0.824 926	0.829 149	0.853 826
CD	0.811 454	0.808 47	0.840 274

3.3.2 参数敏感性分析

为了研究不同特征维度对实验结果的影响,不失一般性,本文选择 $N=1$ 的情况进行分析.通过图 5~7 可以看到,在不同特征维度的情况下,本文提

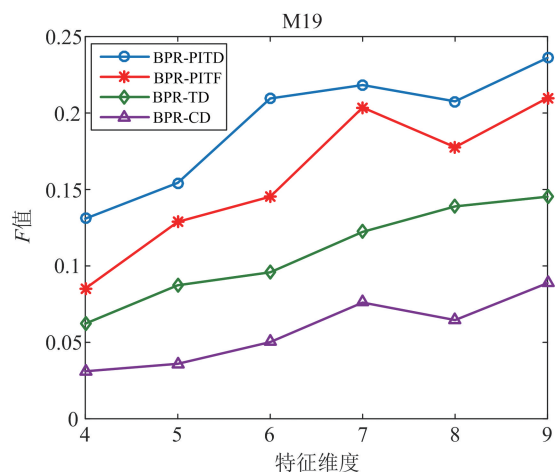
出的 PITD 算法取得最好的推荐效果,而 PITF、TD 和 CD 算法的推荐效果则依次递减.实验中的特征维度对各个算法的性能有较大影响,一般地,较大的特征维度能够获得更好的推荐效果.



实验中选取的特征数为 5.

图 5 M10 数据集上 F 值随特征数的变化趋势

Fig. 5 Change trend of F value with characteristic value on M10 dataset



实验中选取的特征数为 5.

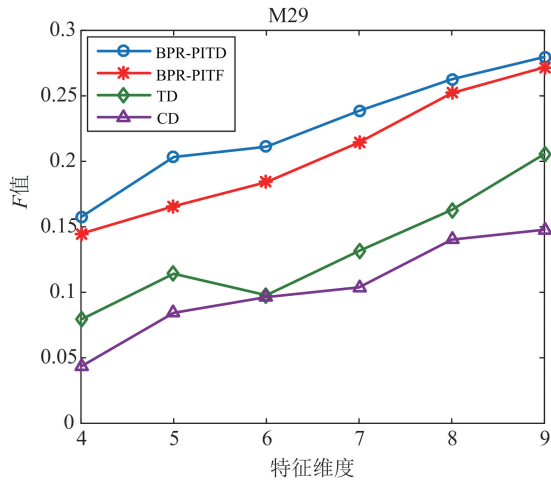
图 6 M19 数据集上 F 值随特征数的变化趋势

Fig. 6 Change trend of F value with characteristic value on M19 dataset

3.3.3 时间开销分析

本文提出的 PITD 模型针对 TD 模型训练和预测需要很高时间复杂度的缺陷做出了相应的改进,为了进一步说明 PITD 在运行时间上的提升,表 4 具体罗列了这两个模型在各数据集上的采样 100 次时的时间开销.从表 4 可以看到,PITD 模型相对 TD 模型显著缩短了运行时间.这是因为 PITD 模型捕捉到了核心张量中的主要部分并且直接忽略了次

要部分,而 TD 模型则是对所有核心张量都进行完整计算。



实验中选取的特征数为 5。

图 7 M29 数据集上 F 值随特征数的变化趋势

Fig. 7 Change trend of F value with characteristic value on M29 dataset

表 4 PITD 与 TD 模型时间开销(单位:s)

Tab. 4 Time consumption of models PITD and TD(Unit: s)

模型	M10 数据集	M19 数据集	M29 数据集
TD	34.11	52.76	81.58
PITD	10.45	15.64	24.29

4 结论

本文研究了标签推荐系统中的难点,并提出了一种基于成对交互张量分解的标签推荐算法 PITD。在此基础上,利用贝叶斯个性化排序 BPR 算法进行推导,并对模型参数进行优化求解,从而得到了完整的标签推荐算法。随后,在真实数据集上进行了大量实验验证,结果表明本文提出的基于 BPR 的 PITD 模型在不同特征维度以及不同推荐指标下几乎都优于对比算法。在未来的工作中,还将进一步考虑用户画像,时间信息对标签推荐的影响。

参考文献(References)

- [1] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C]// Proceedings of the 10th International Conference on World Wide Web. Hong Kong, China: ACM, 2001: 285-295.
- [2] SCHAFFER J, FRANKOWSKI D, HERLOCKER J, et al. Collaborative filtering recommender systems. The adaptive web[M]// The Adaptive Web. Berlin: Springer, 2007, 4321: 291-324.
- [3] SU, X Y, KHOSHGOFTAAR T M. A survey of collaborative filtering techniques [J]. Advances in Artificial Intelligence, 2009, (4): 421425(1-19).
- [4] SALAKHUTDINOV R R, MNIH A. Probabilistic matrix factorization [C]// Proceedings of the 20th International Conference on Neural Information Processing Systems. Vancouver, Canada: ACM, 2008: 1257-1264.
- [5] MA H, YANG H X, LYU M R, et al. SoRec: Social recommendation using probabilistic matrix factorization [C]// Proceedings of the 17th ACM Conference on Information and Knowledge Management. Napa Valley, USA: ACM, 2008: 931-940.
- [6] STEFFEN R, LEANDRO B Y, ALEXANDROS N, et al. Learning optimal ranking with tensor factorization for tag recommendation[C]// Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris: ACM, 2009: 727-736.
- [7] 张斌, 张引, 高克宁, 等. 融合关系与内容分析的社会标签推荐[J]. 软件学报, 2012, 23(3): 476-488.
- [8] WANG H, SHI X J, YEUNG D Y. Relational stacked denoising autoencoder for tag recommendation [C]// Proceedings of the 29th AAAI Conference on Artificial Intelligence. Austin, USA: AAAI, 2015: 3052-3058.
- [9] WANG H, CHEN B Y, LI W J. Collaborative topic regression with social regularization for tag recommendation [C]// Proceedings of the 32th International Joint Conference on Artificial Intelligence. Beijing, China: ACM, 2013: 2719-2725.
- [10] TUAROB S, POUCHARD L C, GILES C L. Automatic tag recommendation for metadata annotation using probabilistic topic modeling[C]// Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries. Indianapolis, USA: ACM, 2013: 239-248.
- [11] ZHONG S R, LEI K, HUANG X H, et al. Topic representation: A novel method of tag recommendation for text [C]// Proceedings of the 2nd International Conference on Big Data Analysis. Beijing, China: IEEE, 2017: 671-676.
- [12] NGUYEN H T H, WISTUBA M, SCHMIDT-THIEME L. Personalized tag recommendation for images using deep transfer learning[C]// Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Macedonia: Springer, 2017: 705-720.
- [13] BELÉM F M, ALMEIDA J M, GONÇALVES M A. A survey on tag recommendation methods. Journal of the Association for Information Science and

- Technology, 2017, 68(4): 830-844.
- [14] NGUYEN H T H, WISTUBA M, GRABOCKA J, et al. Personalized deep learning for tag recommendation [C]// Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. Jeju, South Korea: Springer, 2017: 186-197.
- [15] 刘慧婷, 郭孝雪, 程雷, 等. 基于降噪关系正则化的微博用户标签推荐[J]. 模式识别与人工智能, 2017, 30(10): 907-916.
- [16] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian personalized ranking from implicit feedback [C]// Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence. Montreal, Canada: AUAI Press, 2009: 452-461.
- [17] LAMBIOTTE R, AUSLOOS M. Collaborative tagging as a tripartite network[C]// Proceedings of the International Conference on Computational Science. Berlin: Springer, 2006: 1114-1117.
- [18] PAGE L, BRIN S, MOTWANI R, et al. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab [C]// Proceedings of the WWW Conference. Melbourne, Australia: IEEE, 1998: 161-172.
- [19] HOTH O A, JÄSCHKE R, SCHMITZ C, et al. FolkRank: A ranking algorithm for folksonomies[C]// Proceedings of the FGIR. IEEE, 2006, 1: 111-114.
- [20] SYMEONIDIS P, NANOPOULOS A, MANOLOPOULOS Y. Tag recommendation based on tensor dimensionality reduction [C]// Proceedings of the ACM Conference on Recommender Systems. Lausanne, Switzerland: ACM, 2008: 43-50.
- [21] KOLDA T G, BADER B W. Tensor decompositions and applications [J]. SIAM Review, 2009, 51(3): 455-500.
- [22] RENDLE S, SCHMIDT-THIEME L. Pairwise interaction tensor factorization for personalized tag recommendation [C]// Proceedings of the 3rd ACM International Conference on Web Search and Data Mining. New York: ACM, 2010: 81-90.