·人工智能及识别技术· 文章编号:1000-3428(2018)04-0225-06 文献标志码:A 中图分类号:TP312

基于 SVD ++ 与标签的跨域推荐模型

邢长征,杨晓婷

(辽宁工程技术大学 电子与信息工程学院,辽宁 葫芦岛 125105)

摘 要: 在现有多数跨域推荐模型中,用户不能给指定项目添加标签,并且建立模型时未考虑用户的历史标签,导致推荐误差变大。针对上述问题,构建基于 SVD ++ 模型并融合标签推荐的跨域推荐模型 TagSVD ++。该模型继承 SVD ++ 模型利用评分数据预测的特点,加入用户和项目标签信息,通过标签使用次数反映用户喜好和项目特征,并且引入热门惩罚系数避免热门标签和项目对推荐预测的干扰。在真实电影和图书网站相关数据模拟的跨领域数据集上进行实验,结果表明,TagSVD ++ 模型能有效提高跨域推荐的准确性。

关键词: 跨域推荐;热门惩罚系数;标签推荐;SVD ++ 模型;推荐模型

中文引用格式: 邢长征,杨晓婷. 基于 SVD ++ 与标签的跨域推荐模型[J]. 计算机工程,2018,44(4):225-230.

英文引用格式: XING Changzheng, YANG Xiaoting. Cross-domain Recommendation Model Based on SVD ++ and Tag[J]. Computer Engineering,2018,44(4):225-230.

Cross-domain Recommendation Model Based on SVD ++ and Tag

XING Changzheng, YANG Xiaoting

(School of Electronics and Information Engineering, Liaoning Technical University, Huludao, Liaoning 125105, China)

【Abstract】 Currently, most cross-domain recommendation models have two problems: the prediction error becomes large when the user fails to add a tag to the specified item and when building the model, the user's history label is not taken into account, which causes the decreasing of recommendation result's accuracy. In order to solve these two problems, a cross-domain recommendation model is constructed, named TagSVD ++, which is based on the SVD ++ model and fused with the tag recommendation. This model not only inherits the characteristics of SVD ++ model's using of score data to predict, but also adds the user and the project label information. It uses the tag's frequency to reflect user preferences and characteristics, and introduces the popular penalty factor to prevent hot tags and popular items from bringing interference to recommendation and predictions. Comparative tests on cross-domain data sets for data simulation which is related to real movie and book website are conducted. Experimental results show that TagSVD ++ model can improve the accuracy of cross-domain recommendations effectively.

【Key words】 cross-domain recommendation; hot punish coefficient; tag recommendation; SVD ++ model; recommendation model

DOI: 10.3969/j.issn.1000-3428.2018.04.036

0 概述

推荐系统根据用户的喜好(兴趣和目标)来确定用户感兴趣的潜在项目[1]。大部分的推荐系统都是根据单一领域中的用户和项目历史数据来进行推荐的,例如 Netflix、Last.fm、Barnes & Noble 网站对电影、音乐以及书籍的推荐,但由于用户不仅仅对单一领域的事物感兴趣,因此单领域推荐已经不能准确地预测用户的兴趣。针对单领域推荐模型在数据稀疏和冷启动情况下推荐效果较差的问题,大型电子商务网站 Amazon 和 eBay 通过多个领域来获取用户的反馈信息,从而产生了跨域推荐系统。跨领域推荐系统中的跨域推荐模型[2]是利用不同源数据域来对目标数据域[3-4]信息进行预测的。现如今各大网站利用标签信息来体现用户喜好和项目特征,而用户在给不同领域(如电影、音乐、书籍等领域)的项目添加标签时,可以使用通用词汇作为标签[5]。因此,标签可以作为连接不同领域的桥梁并且用来克服基于内容推荐时特征信息复杂性和不均匀性的不足[6-7]。目前,将跨域推荐模型融入标签因素来进行评分预测时,仍存在以下问题:UserItemTags 模型[8]当用户没有给特定项目添加标签时预测误差变大,

作者简介:邢长征(1967—),男,教授,主研方向为数据挖掘;杨晓婷(通信作者),硕士研究生。

收稿日期:2017-02-09 修回日期:2017-03-26 E-mail:L_tt0628@163.com

ItemRelTags[8]只是加入了项目的标签信息,并没有考虑用户标签信息,导致无法利用用户过去选择的标签来预测用户的喜好。

为提高预测准确度,本文构建一个新的跨域推荐模型。在使用 SVD ++ 模型评分数据预测的基础上,添加用户和项目历史标签信息,并利用标签使用次数、热门标签和项目的惩罚系数刻画并预测用户喜好。

1 相关研究

1.1 基于矩阵分解的推荐模型

1.1.1 隐因式分解模型

隐因式分解模型(Latent Factor Model,LFM)是推荐系统领域的热点研究话题之一,其核心思想是通过隐含特征联系用户兴趣和物品[9]。具体过程分为 3 个部分:将物品映射到隐含分类,确定用户对隐含分类的兴趣,选择用户感兴趣的分类中的物品推荐给用户[10]。LFM 通过式(1)计算用户 u 对物品 i 的兴趣:

$$\hat{r}_{u,i}=\sum_{k=1}^{K}p_{u,k}q_{i,k} \tag{1}$$

其中,$p_{u,k}$和 $q_{i,k}$是模型的参数,$p_{u,k}$代表用户 u 的兴趣是第 k 个隐类的权重,$q_{i,k}$代表第 k 个隐类属于物品 i 的权重,K 是隐类的个数。两者的乘积再对 k 积分就得到用户 u 对商品 i 的偏好程度。

将式(1)表示为向量乘积的形式,即为:

$$\hat{r}_{u,i}=\boldsymbol{p}_u^{\mathrm{T}}\boldsymbol{q}_i \tag{2}$$

考虑到在真实的模型中,每个用户评分的基准线不同,每个商品得到评分的基准线也不同,引入参数 μ 表示训练集中所有记录的评分的全局平均数;引入参数 b_u,即用户偏置项,这一项表示用户的评分习惯中和物品没有关系的那种因素;引入参数 b_i,即物品偏置项,表示物品接受的评分中和用户没有什么关系的因素。因此,式(2)变为以下形式:

$$\hat{r}_{u,i}=\mu+b_u+b_i+\boldsymbol{p}_u^{\mathrm{T}}\boldsymbol{q}_i \tag{3}$$

1.1.2 SVD ++ 模型

LFM 模型中并没有显式地考虑用户的历史行为对用户评分预测的影响,为此 Koren 在 Netflix Prize 比赛中构建一个模型,将用户历史评分的物品加入到 LFM 模型中。该模型被称为 SVD ++[10-12],具体如下:

将基于项目的协同过滤算法(ItemCF)[13]的预测计算式改为:

$$\hat{r}_{u,i}=\frac{1}{\sqrt{|N(u)|}}\sum_{j\in N(u)}\boldsymbol{w}_{i,j} \tag{4}$$

其中,$N(u)$ 是用户 u 提供隐式偏好的项目集。式(4)的缺点为物品相似度矩阵 $\boldsymbol{w}_{i,j}$是一个稠密矩阵,存储它需要比较大的空间。因此,将 $\boldsymbol{w}_{i,j}$矩阵分解为向量乘积的形式,即 $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{y}_j$,模型如下:

$$\hat{r}_{u,i}=\frac{1}{\sqrt{|N(u)|}}\sum_{j\in N(u)}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{y}_j=\frac{1}{\sqrt{|N(u)|}}\boldsymbol{x}_i^{\mathrm{T}}\sum_{j\in N(u)}\boldsymbol{y}_j \tag{5}$$

将前面的隐因式分解模型(LFM)和上面的模型相加,从而得到:

$$\hat{r}_{u,i}=\mu+b_u+b_i+\boldsymbol{p}_u^{\mathrm{T}}\boldsymbol{q}_i+\frac{1}{\sqrt{|N(u)|}}\boldsymbol{x}_i^{\mathrm{T}}\sum_{j\in N(u)}\boldsymbol{y}_j \tag{6}$$

文献[10]提出为了不避免太多参数造成过拟合,可以令 $x=q$,从而得到最终的 SVD ++ 模型,用式(7)表示 SVD ++ 模型:

$$\hat{r}_{u,i}=\mu+b_u+b_i+\boldsymbol{q}_i^{\mathrm{T}}\left(\boldsymbol{p}_u+\frac{1}{\sqrt{|N(u)|}}\sum_{j\in N(u)}\boldsymbol{y}_j\right) \tag{7}$$

1.2 基于标签的跨域推荐模型

用户使用社交标签对信息进行分类[14],可自由组织、管理和搜索所需的资源。这种易使用性使标签成为信息分类与索引的重要方式,既能反映出用户的兴趣爱好,又能体现资源特征,并且对稀疏数据和新资源推荐质量的提升有所帮助。在跨域推荐过程中,在其中一个领域中使用的标签,完全可以被重新用于另一个领域[15]。因此,可以在跨域推荐模型中利用标签信息作为连接不同领域的桥梁。

1.2.1 UserItemTags 模型

UserItemTags[8]模型是一个通过标签体现用户兴趣偏好和资源特征信息的模型,在预测评分时利用了目标用户使用过的标签。使用此模型时假设用户没有对项目进行评分而只是添加了标签,然后利用这些标签来更好地预测用户对项目的评分。运用此模型进行推荐的一个经典案例是 Delicious 的社交书签网站,用户可以标记书签,但可以不对项目进行评分[8]。UserItemTags 使用下式预测评分:

$$\hat{r}_{u,i}=\boldsymbol{p}_u^{\mathrm{T}}\left(\boldsymbol{q}_i+\frac{1}{|T_u(i)|}\sum_{t\in T_u(i)}\boldsymbol{y}_t\right) \tag{8}$$

其中,T_u 表示用户 u 给项目 i 添加的标签集合。

UserItemTags 模型缺点为:当项目没有被用户添加标签时推荐预测性能会大幅度降低。

1.2.2 ItemRelTags 模型

ItemRelTags[8]模型中没有涉及目标用户对目标项目添加的标签,而是使用了任意用户对目标项目添加的所有标签,这样就可以克服 UserItemTags 的局限性:目标用户在没有对某一项目添加标签的情况下无法进行评分预测。事实上,ItemRelTags 模型只需要所有用户对项目 i 添加有关标签集合 $TR(i)$。这里评估标签的相关性使用 Wilcoxon 秩和检验方法来决定每个标签的去留[8]。此外,假设标签对于

任何用户都具有相同影响。由于在模型中用户给目标项目添加相关标签,因此相同的标记会出现多次。如果一个项目被添加同一个标签很多次,这就说明这个标签可以很好地反映此项目的特征。因此,该模型添加了标签的使用频率 n_t。ItemRelTags 模型的计算式如下:

$$\hat{r}_{u,i}=\boldsymbol{p}_u^{\mathrm{T}}\left(\boldsymbol{q}_i+\frac{1}{|TR(i)|}\sum_{t\in TR(i)}n_t\boldsymbol{y}_t\right) \tag{9}$$

其中,$TR(i)$ 是整个标签集合中和项目 i 有关的标签,n_t 是用户给项目 i 添加标签 t 的次数。该模型的优点是即使是没有对项目做任何标签的新用户也可以为其做出评分预测。但其只是考虑特定项目的历史标签并没有考虑目标用户的历史标签,在这种情况下,可以通过用户在过去选择的标签来丰富用户的喜好。

2　TagSVD ++ 跨域推荐模型

上文介绍的隐因式分解和 SVD ++ 模型虽然在推荐领域中流行度较高,但只能在单一领域中进行推荐,这样就会在冷启动情况下出现推荐准确度下降的问题。根据用户不只对单一领域感兴趣的现状,本文构建跨域推荐模型,用不同领域共有的标签信息作为连接不同领域的纽带,在多个领域信息中进行推荐预测,提升了预测的性能[16]。目前跨域推荐模型存在的不足是:利用 UserItemTags 模型进行预测,用户没有为项目添加标签时,模型预测效果没有原始模型隐因式分解(LFM)模型预测精准;ItemRelTags 模型并没有考虑特定用户的历史标签来丰富用户的喜好。

本文构建基于 SVD ++ 改进标签跨域推荐模型 TagSVD ++。该模型继承了 SVD ++ 中利用评分信息间接表示用户喜好和项目特征的特点,同时又加入了用户以及项目的历史标签信息和标签使用频率,并通过热门标签惩罚系数来增强模型推荐效果的准确性和新颖性。

步骤 1　求相似的标签。

针对新用户或者新物品标签集合中标签数量较少的情况,需要对标签集合进行扩展,标签扩展的本质是对每个标签找到和它相似的标签,也就是计算标签之间的相似度。首先通过式(10)余弦相似度公式计算标签 b 和 b'之间的相似度:

$$sim(b,b')=\frac{\sum_{i\in N(b)\cap N(b')}n_{b,j}n_{b',j}}{\sqrt{\sum_{i\in N(b)}n_{b,j}^2\sum_{i\in N(b')}n_{b',j}^2}} \tag{10}$$

然后对标签集合排序,参考文献[17]中将排序结果中前 20 个标签作为用户相关的标签。

步骤 2　构造模型。

本文模型构建过程如图 1 所示。

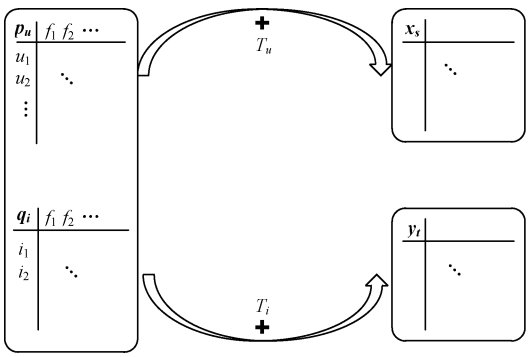


图 1　模型构建过程

模型公式表示为:

$$\hat{r}_{u,i}=\left(\boldsymbol{p}_u+\frac{1}{|T_u|}\sum_{s\in T_u}\frac{n_{u,s}\boldsymbol{x}_s}{\ln(1+\boldsymbol{n}_s^{(u)})}\right)^{\mathrm{T}}\cdot\left(\boldsymbol{q}_i+\frac{1}{|T_i|}\sum_{t\in T_i}\frac{n_{i,t}\boldsymbol{y}_t}{\ln(1+n_i^{(u)})\ln(1+n_t^{(u)})}\right) \tag{11}$$

其中,$\hat{r}_{u,i}$为用户预测评分,p_u 和 q_i 是继承于 SVD ++ 模型中的参数,间接地通过评分数据来对用户喜好进行预测,而在原始 SVD ++ 模型中 $\boldsymbol{y}$ 向量用来间接表示用户的评分信息,应用到 TagSVD ++ 模型中用 $\boldsymbol{y}_t$ 来间接表示项目标签信息,并添加了向量 $\boldsymbol{x}_s$ 用来间接地表示用户标签信息。T_u 是用户 u 给任意项目添加的标签的集合,T_i 是项目 i 被任意用户添加的标签的集合。$n_{u,s}$是被用户 u 添加标签 s 的项目数量,$n_{i,t}$是把标签 t 添加给项目 i 的用户数量。$n_s^{(u)}$、$n_t^{(u)}$ 分别记录了标签 s、t 被多少不同的用户使用过,$n_i^{(u)}$ 记录了物品 i 被多少个不同的用户打过标签,$\ln(1+n_s^{(u)})$、$\ln(1+n_t^{(u)})$和 $\ln(1+n_i^{(u)})$分别是热门标签和热门商品的惩罚系数。因为$|T_u|=\sum_{s\in T_u}n_{u,s}$ $|T_i|=\sum_{s\in T_i}n_{i,t}$,所以 $\frac{n_{i,t}}{|T_i|\ln(1+n_i^{(u)})\ln(1+n_t^{(u)})}<1$,$\frac{n_{u,s}}{|T_u|\ln(1+n_s^{(u)})}<1$,即 $\boldsymbol{p}_u$、$\boldsymbol{q}_i$ 代表的评分因素仍对模型的预测占主导位置。

在整个模型中 $\boldsymbol{p}_u$、$\boldsymbol{q}_i$ 的作用是利用评分数据来进行推荐预测,而$\frac{1}{|T_u|}\sum_{s\in T_u}\boldsymbol{x}_s$ 的作用就是充分利用用户历史标签来预测用户的喜好,此处补充了 ItemRelTags 模型没有考虑的部分:特定用户对目标项目添加的标签集合。同样,$\frac{1}{|T_i|}\sum_{s\in T_i}\boldsymbol{y}_t$ 的作用是利用项目被添加过的标签来推测项目的属性。$n_{u,s}$、$n_{i,t}$的加入体现了标签的使用频率,更好地利用标签来描述用户的喜好,热门标签和项目的惩罚系数则是为了防止推荐结果一味地偏向热门标签和项目,为推荐预测增加了新颖性。

步骤 3 损失函数计算。

类似于 SVD ++ 模型,需要求得式(11),通过用户实际评分和预测评分之间损失函数的最小值来找到最合适的参数。

$$c=\min_{p,q,x,y}\sum_{(u,i)\in\mathbb{R}}\frac{1}{2}(r_{u,i}-\hat{r}_{u,i})^2+\frac{\lambda}{2}(||\boldsymbol{p}_u||^2+||\boldsymbol{q}_i||^2+\sum_{s\in T_u}||\boldsymbol{x}_t||^2+\sum_{s\in T_i}||\boldsymbol{y}_t||^2) \tag{12}$$

最小误差函数中的$\frac{1}{2}$只是为了简化系数,对结果并无影响,$\frac{\lambda}{2}(||p_u||^2+||q_i||^2+\sum_{s\in T_u}||\boldsymbol{x}_t||^2+\sum_{s\in T_i}||\boldsymbol{y}_t||^2)$是用来防止过拟合的正则化项。

步骤 4 使用随机梯度下降法[18]求损失函数最小值。

最小化上述的损失函数,利用随机梯度下降模型来计算。该模型是最优化理论里最基础的优化模型,通过求参数的偏导数找到最速下降的方向,然后通过迭代法不断地优化参数。根据随机梯度下降法,需要将参数沿着最速下降方向向前推进,因此,得到如式(13)~式(16)所示的递推公式。

$$\boldsymbol{p}_u\leftarrow\boldsymbol{p}_u-\alpha(\lambda\boldsymbol{p}_u-\frac{\partial c}{\partial\boldsymbol{p}_u}) \tag{13}$$

$$\boldsymbol{q}_i\leftarrow\boldsymbol{q}_i-\alpha(\lambda\boldsymbol{q}_i-\frac{\partial c}{\partial\boldsymbol{q}_i}) \tag{14}$$

$$\boldsymbol{x}_a\leftarrow\boldsymbol{x}_a-\alpha(\lambda\boldsymbol{x}_a-\frac{\partial c}{\partial\boldsymbol{x}_a}),\forall a\in T_u \tag{15}$$

$$\boldsymbol{y}_b\leftarrow\boldsymbol{y}_b-\alpha(\lambda\boldsymbol{y}_b-\frac{\partial c}{\partial\boldsymbol{y}_b}),\forall b\in T_i \tag{16}$$

其中,α 代表学习速率,λ 代表正则化参数,$\frac{\partial c}{\partial\boldsymbol{p}_u}$、$\frac{\partial c}{\partial\boldsymbol{q}_i}$、$\frac{\partial c}{\partial\boldsymbol{x}_a}$、$\frac{\partial c}{\partial\boldsymbol{y}_b}$如式(17)~式(20)所示。

$$\frac{\partial c}{\partial\boldsymbol{p}_u}=-(r_{u,i}-\hat{r}_{u,i})\cdot\left(\boldsymbol{q}_i+\frac{1}{|T_i|}\sum_{s\in T_i}\frac{n_{i,t}\boldsymbol{y}_t}{\ln(1+n_i^{(u)})\ln(1+n_t^{(u)})}\right)+\lambda\boldsymbol{p}_u \tag{17}$$

$$\frac{\partial c}{\partial\boldsymbol{q}_i}=-(r_{u,i}-\hat{r}_{u,i})\left(\boldsymbol{p}_u+\frac{1}{|T_u|}\sum_{s\in T_u}\frac{n_{u,s}\boldsymbol{x}_s}{\ln(1+n_s^{(u)})}\right)+\lambda\boldsymbol{q}_i \tag{18}$$

$$\frac{\partial c}{\partial x_a}=-(r_{u,i}-\hat{r}_{u,i})\frac{1}{|T_u|}\cdot\frac{n_{u,a}}{\ln(1+n_s^{(u)})}\cdot\left(\boldsymbol{q}_i+\frac{1}{|T_i|}\sum_{s\in T_i}\frac{n_{i,t}\boldsymbol{y}_t}{\ln(1+n_i^{(u)})\ln(1+n_t^{(u)})}\right)+\lambda x_a,\forall a\in T_u \tag{19}$$

$$\frac{\partial c}{\partial y_b}=-(r_{u,i}-\hat{r}_{u,i})\frac{1}{|T_i|}\cdot\frac{n_{i,b}}{\ln(1+n_i^{(u)})\ln(1+n_t^{(u)})}\left(\boldsymbol{p}_u+\frac{1}{|T_u|}\sum_{s\in T_u}\frac{n_{u,s}\boldsymbol{x}_s}{\ln(1+n_s^{(u)})}\right)+\lambda y_b,\forall b\in T_i \tag{20}$$

利用递推公式进行迭代直至值不再发生变化则迭代结束,从而利用随机梯度下降法求得使损失函数最小的各个参数:学习速率 α 和正则化参数 λ。表 1 列出了跨域模型对标签信息的利用情况,其中√代表包括,×表示不包括。

表 1 跨域模型比较

模型	利用项目标签推荐	利用用户标签推荐	利用标签使用频率优化推荐	添加热门标签项目惩罚系数
UserItemTags	√	×	×	×
ItemReltags	√	×	√	×
TagSVD ++	√	√	√	√

3 实验

3.1 实验数据及预处理

3.1.1 数据集

为了模拟跨域数据集,本文使用 2 个来自不同领域公开可下载的数据集:MovieLens 数据集和 LibraryThings 数据集。2 个数据集评分范围均为 1 分~5 分,每 0.5 分为一级,评分代表用户的偏好程度。实验数据集规模如表 2 所示,原始数据集的典型示例如表 3 和表 4 所示。

表 2 实验数据集规模

数据集	用户数量	项目数量	评分数量	标签数量	共同标签数量
LibraryThings	204	1 305	24 573	4 623	1 334
MovieLens	2 018	5 017	24 573	9 488	

表 3 MovieLens 数据集典型示例

用户编号	项目编号	评分	标签
1107	3441	Null	fiction
1778	2357	Null	love
33770	59725	4.0	NewYorkCity
71262	2478	Null	funny movie
71254	54272	Null	funny

表 4 LibraryThings 原始数据集典型示例

用户编号	项目编号	评分	标签
1	7391	2.5	fiction
7	10104	5.0	london
4321	1729243	3.5	fiction
20	675	Null	non-fiction
2	1710152	Null	nonfiction
1	3067	4.0	love

3.1.2　标签剪修

对标签进行修剪:1)去除用户对项目添加的表示情绪的标签,比如“不好笑”,但是“不好笑”不能认为是用户的兴趣;2)去除词频很高的停止词;3)去除因词根不同造成的同义词;4)去除因分隔符造成的同义词。

3.2　评估方法

评估模型模型时,首先将 MovieLens 作为源数据域,LibraryThings 作为目标数据域,然后反之也是如此。将目标数据域平均分为没有重叠的 10 份,在每一份数据中,10% 的数据为测试集来评估方法的性能。剩下的 90% 数据做为训练数据集,在训练数据集中有 20% 的验证数据集,用来寻找模型的最优参数,其中包括学习速率 α、正则化参数 λ 和隐特征个数 k。而训练数据中剩下的 80% 数据结合源数据域来建立模型。为了测评目标数据域中用户评分的稀疏程度对模型性能的影响,将训练数据平均分为 10 份,依次取 1 份,2 份,……,10 份作为训练数据来训练模型以此来模拟用户评分数据不同的稀疏程度[15]。因为数据集中整个评分系统是以0.5 分为一等级建立的,所以采用平均绝对误差(Mean Absolute Error,MAE)即来进行预测可以降低预测结果的误差[15]。MAE 采用绝对值计算预测误差,其定义为:

$$M_{\mathrm{MAE}}=\frac{\sum_{u,i\in T}|r_{u,i}-\hat{r}_{u,i}|}{|T|} \tag{21}$$

3.3　实验结果与分析

各模型的平均最优参数如表 5 和表 6 所示。

表 5　LibraryThings(源数据域)各模型平均最优参数

模型	α	k	λ
LFM	0.020	50	0.010
SVD ++	0.020	41	0.007
UserItemTags	0.019	46	0.003
ItemRelTags	0.017	40	0.001
TagSVD ++	0.015	40	0.030

表 6　MovieLens(源数据域)各模型平均最优参数

模型	α	k	λ
LFM	0.020	46	0.010
SVD ++	0.020	41	0.006
UserItemTags	0.019	46	0.010
ItemRelTags	0.020	46	0.006
TagSVD ++	0.020	45	0.040

将 LibraryThings 数据集作为源数据集,将 MovieLens 数据集作为目标数据集,实验结果如图 2 所示。可以看出,LFM、SVD ++ 单领域推荐模型的误差远远大于跨领域的推荐模型,而在跨域推荐模型中 TagSVD ++ 模型的预测误差都远小于其他几个模型,即使只有 10% 的评分数据(冷启动)存在的情况下。随着目标域评分量的增多平均绝对误差也在不断变小。

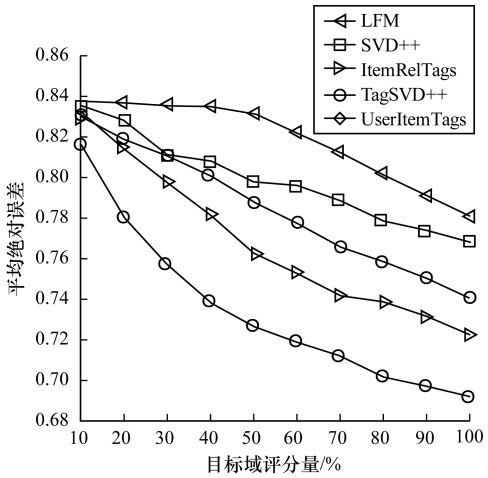


图 2　平均绝对误差比较 1

将 MovieLens 数据集作为源数据集,将 LibraryThings 数据集作为目标数据集,实验结果如图 3所示。与前一种情况类似,跨域推荐模型推荐误差远远小于单域推荐模型,而在跨域推荐模型中 TagSVD ++ 推荐准确度又远远高于另外 4 个模型。而由图 2 和图 1 可以看出,第2 种情况下所有的模型的预测误差都大于第 1 种情况,这是因为 MovieLens 数据域数据量大于 LibraryThings 数据域的数据量。因此,较少的训练数据会导致预测精准度的下降。

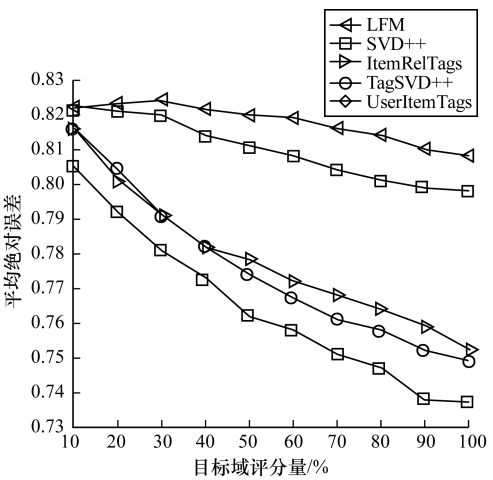


图 3　平均绝对误差比较 2

经过 20 次重复试验求得到的每组结果的标准差,如表 7 和表 8 所示。可以看出,TagSVD ++ 在各个数据稀疏度下标准差都小于其他模型。这说明 TagSVD ++ 模型的预测结果不仅误差较其他算法小,而且预测结果稳定,可以被用来很好地预测用户的评分。

表 7 LibraryThings(源数据域)各模型标准差

模型	10% 稀疏度	30% 稀疏度	50% 稀疏度	70% 稀疏度	90% 稀疏度
LFM	0.838 ±0.030	0.836 ±0.019	0.831 ±0.014	0.812 ±0.022	0.791 ±0.024
SVD ++	0.836 ±0.028	0.811 ±0.017	0.798 ±0.012	0.789 ±0.020	0.774 ±0.024
ItemRelTags	0.832 ±0.027	0.798 ±0.015	0.762 ±0.012	0.742 ±0.020	0.731 ±0.022
TagSVD ++	0.816 ±0.026	0.758 ±0.013	0.727 ±0.013	0.712 ±0.018	0.697 ±0.020

表 8 MovieLens(源数据域)各模型标准差

模型	10% 稀疏度	30% 稀疏度	50% 稀疏度	70% 稀疏度	90% 稀疏度
LFM	0.822 ±0.024	0.824 ±0.017	0.820 ±0.021	0.816 ±0.016	0.810 ±0.020
SVD ++	0.822 ±0.021	0.820 ±0.015	0.811 ±0.017	0.804 ±0.013	0.799 ±0.016
ItemRelTags	0.816 ±0.019	0.791 ±0,015	0.778 ±0.014	0.768 ±0.012	0.759 ±0.015
TagSVD ++	0.805 ±0.017	0.781 ±0.012	0.762 ±0.013	0.751 ±0.011	0.738 ±0.012

4 结束语

目前利用标签信息来连接 2 个不同领域已经成为一种可靠有效的方式,即使在冷启动的情况下,本文建立的 TagSVD ++ 模型也能把源数据域中的信息更有效地传递给目标域。实验结果表明,该模型利用标签信息与用户和项目的融合,可有效提高用户兴趣和项目特征的预测精度。下一步将在不同领域中存在相同标签数量较少或者没有的条件下连接 2 个领域,并研究不同领域中相同标签存在语义差异的情况。

参考文献

[1] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender systems: a survey of the state of-the-art and possible [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6): 734-749.

[2] LI B. Cross-domain collaborative filtering: a brief survey[C]// Proceedings of IEEE International Conference on Tools with Artificial Intelligence. Washington D. C., USA: IEEE Press, 2011: 1085-1086.

[3] GAO S, LUO H, CHEN D, et al. Cross-domain recommendation via cluster-level latent factor model [C]// Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases. Berlin, Germany: Springer, 2013: 161-176.

[4] LI B, YANG Q, XUE X. Can movies and books collaborate? cross-domain collaborative for sparsity reduction [C]//Proceedings of the 21st International Joint Conference on Artificial Intelligence. Berlin, Germany: Springer, 2009: 2052-2057.

[5] CREMONESI P, TRIPODI A, TURRIN R. Cross-domain recommender systems [C]//Proceedings of IEEE International Conference on Data Mining Workshops. Washington D. C., USA: IEEE Press, 2011: 496-503.

[6] ENRICH M, BRAUNHOFER M, RICCI F. Cold-start management with cross-domain collaborative filtering and tags [C]//Proceedings of the 14th International Conference on E-commerce and Web Technologies. Berlin, Germany: Springer, 2013: 101-112.

[7] SHI Y, LARSON M, HANJALIC A. Tags as bridges between domains: improving recommendation with tag induced cross-domain collaborative filtering [C]// Proceedings of the 19th International Conference on User Modeling Adaption and Personalization. Berlin, Germany: Springer, 2011: 305-316.

[8] ENRICH M, BRAUNHOFER M, RICCI F. Cold-start management with cross-domain collaborative filtering and tags[M]. Berlin, Germany: Springer, 2014: 36-37.

[9] FUNK S. Netflix update: try this at home [EB/OL]. (2006-12-11). http://sifter.org/~simon/journal/20061211.html.

[10] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems [J]. IEEE Computer, 2009, 43(8): 30-37.

[11] KOREN Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C]// Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2008: 426-434.

[12] KOREN Y, BELL R. Advances in collaborative filtering[M]//RICCI F, ROKACH L, SHAPIRA B. Recommender Systems Handbook. Berlin, Germany: Springer, 2011: 145-186.

[13] GREG L, BRENT S, JEREMY Y. Amazon.com recommendations: item-to-item collaborative filtering [J]. IEEE Internet Computing, 2003, 7(1): 76-79.

[14] 张 斌,张 引,高克宁. 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.

[15] CANTADOR F I. Cross-domain recommender systems: a survey of the state of the art[C]//Proceedings of the 2nd Spanish Conference on Information Retrieval. Washington D. C., USA: IEEE Press, 2012: 187-198.

[16] SHI Y, LARSON M, HANJALIC A. Exploiting social tags for cross-domain collaborative filtering [C]// Proceedings of UMAP'11. Berlin, Germany: Springer, 2011: 2-18.

[17] 项 亮. 推荐系统实践[M]. 北京: 人民邮电出版社, 2012: 108-109.

[18] WANG B B, WANG Y X. Some properties relating to stochastic gradient descent methods [J]. Journal of Mathematics, 2011, 31(6): 1041-1044.

编辑 金胡考