DOI: 10.3785/j.issn.1008-973X.2015.07.017

基于结构投影非负矩阵分解的协同过滤算法

居 斌[1,2],钱沄涛[1],叶敏超[1]
(1.浙江大学 计算机学院,浙江 杭州 310027;2.浙江省卫生信息中心,浙江 杭州 310006)

摘 要:针对在协同过滤算法中,传统矩阵分解技术在降维过程中会破坏数据相邻结构的问题,提出基于结构投影非负矩阵分解的协同过滤算法(CF-SPNMF).该算法包含离线学习和在线搜索 2 个阶段.在离线学习阶段,通过对用户评分矩阵的投影非负矩阵分解,同时保留用户特征的聚类结构,得到低维的用户潜在兴趣因子.在线搜索阶段,将用户潜在兴趣因子进行余弦相似性匹配,发现目标用户与训练样本用户之间兴趣最相似的邻域集合.在实际数据集上的实验结果表明,提出的 CF-SPNMF 算法与单纯使用矩阵分解和单纯在原评分矩阵上进行用户聚类的推荐算法相比,能够更有效地预测用户实际评分.

关键词:协同过滤;投影非负矩阵分解;相邻结构;聚类
中图分类号:TP 181 **文献标志码**:A **文章编号**:1008-973X(2015)07-1319-07

Collaborative filtering algorithm based on structured projective nonnegative matrix factorization

JU Bin[1,2], QIAN Yun-tao[1], YE Min-chao[1]
(1. *College of Computer Science, Zhejiang University, Hangzhou* 310027, *China*;
2. *Health Information Center of Zhejiang Province, Hangzhou* 310006, *China*)

Abstract: In collaborative filtering algorithm, the classical matrix factorization may destroy the adjacent structures among data points from high dimension to low dimension. A novel collaborative filtering algorithm based on structured projective nonnegative matrix factorization (CF-SPNMF) was proposed in order to overcome the problem. The algorithm contains both offline learning and online searching. In the offline learning stage, projective nonnegative matrix factorization was applied to obtain the low dimensional latent factors of user preference without changing the intrinsic structure of users' cluster. In the online searching stage, cosine similarity was used to measure the similarity between the target user and training users based on the latent factors inferred in the offline stage. Then the most similar neighbor set was further found. The extensive experiments on real-world data set demonstrate that the proposed CF-SPNMF achieves better rating prediction performance than traditional methods using either matrix factorization or users clustering in original rating matrix.

Key words: collaborative filtering; projective nonnegative matrix factorization; adjacent structure; clustering

协同过滤(collaborative filtering,CF)算法是产品推荐系统的主流方法[1-2],主要有基于记忆的协同推荐[3]和基于模型的协同推荐[4].它要解决的问题如下:已知用户对少量物品的评分记录(一般为 1~

收稿日期:2014-04-14. **浙江大学学报(工学版)网址**:www.journals.zju.edu.cn/eng
基金项目:浙江省自然科学基金资助项目(Y1101359);国家科技支撑计划资助项目(2011BAD24B03).
作者简介:居斌(1975-),男,博士生,从事机器学习、数据挖掘的研究.ORCID:0000-0003-4709-4297.E-mail:jubin_bz@163.com
通信联系人:钱沄涛,男,教授,博导.ORCID:0000-0002-5267-239X. E-mail: ytqian@zju.edu.cn

5分),预测用户对其他物品的评分.

传统的基于用户相似性或物品相似性的协同推荐算法主要有2个问题:1)评分表非常庞大且稀疏,缺失的评分项容易造成相似性计算不准确,预测精度会随之下降.2)算法的可扩展性不够,对于新增用户都要在整个评分矩阵中重新搜索一遍,成为推荐系统的性能瓶颈之一.如果把预测评分矩阵问题看成是基于矩阵分解技术求解矩阵补全问题,那么矩阵分解技术能够有效地解决数据高维问题.事实上,矩阵分解已成为CF领域的研究热点[5].

奇异值分解(singular value decomposition, SVD)是常用的矩阵分解方法,给出了评分矩阵的低秩逼近最优解的解析表示[6],通过folding-in投影方法可以实现SVD的增量分解,从而对大规模矩阵分解获得较好的性能[7].由于评分矩阵是高度稀疏,SVD无法给出理想的低秩分解结果[8].概率因子矩阵分解PMF具有对稀疏数据适应性好、预测精度高的优点,目前成为CF中的矩阵分解主流方法[9].它存在如下2个问题:1)降维过程中会破坏数据相邻结构;2)对评分矩阵新加入的用户需要重新计算矩阵分解,不适合互联网应用的在线计算.

为了克服传统矩阵分解技术存在的问题,在不增加除评分矩阵之外的先验信息情况下,本文提出基于结构投影非负矩阵分解(structured projective nonnegative matrix factorization, SPNMF)的协同过滤算法.该算法包含离线学习和在线搜索2个阶段.在离线学习阶段,通过对用户评分矩阵的投影非负矩阵分解,同时保留用户特征的聚类结构,从而得到低维的用户潜在兴趣因子.在线搜索阶段,将用户潜在兴趣因子进行余弦相似性匹配,由此发现目标用户与训练样本用户之间兴趣最相似的邻域集合,从而获得一种高效高精度的推荐算法.

1 结构投影非负矩阵分解

1.1 模型概述

矩阵分解的基本假设是,用户选择物品的兴趣总是分布在低维分类d中,这些分类可以是物品的类别,也可以是潜在购物兴趣等,因此将原评分矩阵$\boldsymbol{X}_{m\times n}$分解为$\boldsymbol{W}_{m\times d}$和$\boldsymbol{H}_{d\times n}$两个低秩矩阵,然后通过$\boldsymbol{X}\approx\boldsymbol{WH}$重构原矩阵,获得其他缺失的评分记录.

非负矩阵分解(nonnegative matrix factorization, NMF)具有"整体由部分组成"的哲学观点[10],使得分解后的矩阵潜在因子更具有可解释性.在协同过滤问题中,给定一个非负数据矩阵$\boldsymbol{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_N]\in\mathbf{R}^{M\times N}$,即$N$个用户对$M$个物品的评分矩阵,$\boldsymbol{X}_{ij}$表示第$j$个用户对第$i$个物品的评分,$\boldsymbol{X}$的每一列是用户评分空间的数据点.非负矩阵分解(NMF)的目标函数为

$$\min\|\boldsymbol{X}-\boldsymbol{WH}\|^2;\boldsymbol{W}\geqslant 0,\boldsymbol{H}\geqslant 0. \tag{1}$$

式中:$\|\cdot\|$指Frobenius范数.式(1)分解后的$\boldsymbol{W}_{m\times d}$矩阵($d<N$)表示用户兴趣特征的基向量($w_1,\cdots,w_d$)构成的矩阵,$\boldsymbol{H}_{d\times n}$矩阵的每一列($h_1,\cdots,h_n$)包含了这组基向量的非负线性组合系数,NMF非负性更是符合用户兴趣因子可加性的特征.

NMF通常没有唯一解,Hoyer[11]指出NMF分解得到的基向量越稀疏,学习到的特征局部性效果越好.胡俐蕊等[12]指出在非负性的约束下,基向量越正交,基向量的稀疏性越好.Yuan等[13]提出投影非负矩阵分解(PNMF)模型,并指出PNMF的W基矩阵正交性和稀疏性都比普通NMF好,PNMF的目标函数如下式所示:

$$\min\|\boldsymbol{X}-\boldsymbol{WPX}\|^2;\boldsymbol{W}\geqslant 0. \tag{2}$$

若令$\boldsymbol{H}=\boldsymbol{PX}$,则表明可以通过适当的线性变换$\boldsymbol{P}$将用户评分数据$\boldsymbol{X}$变换成在特征基矩阵$\boldsymbol{W}$下的系数矩阵$\boldsymbol{H}$.

现有NMF及扩展算法对新增用户(即评分矩阵中的列向量$\boldsymbol{x}_{\text{new}}$)都需要重新计算矩阵分解,非常耗时,不适合互联网在线计算.若采用PNMF算法,则对于新加的用户评分向量$\boldsymbol{x}_{\text{new}}$,相应的系数编码向量$\boldsymbol{h}_{\text{new}}$可由下式计算:

$$\boldsymbol{h}_{\text{new}}=\boldsymbol{P}\boldsymbol{x}_{\text{new}}. \tag{3}$$

然后$\boldsymbol{h}_{\text{new}}$与$\boldsymbol{H}$系数矩阵的每列进行低维特征空间上的用户相似度计算,由此获得评分矩阵缺失项的预测评分.

单纯运用PNMF降维会带来数据点在低维空间重叠或偏离,导致相似性计算受到影响.比如,仿真一个评分数据在3维空间上的分布(见图1),若按主成分分析(PCA)算法在2维空间上进行投影,部分数据点在2维空间重叠在一起,则总体上没有呈现明显的聚类效果(见图2).为了修正PNMF降维过程中忽视原数据分布结构的问题,Yang等[14-15]利用流形结构中所谓的intrinsic graph和penalty graph,分别对PNMF和NMF施加正则化约束.intrinsic graph表示高维空间的相近数据点降维之后应该尽量最小化它们的距离,penalty graph表示高维空间不同类别的数据点之间降维之后应该尽量最大化它们之间的距离.因为penalty graph要用到类别信息,该模型属于有监督的降维.

笔者从"用户A和B如果在高维的评分空间如

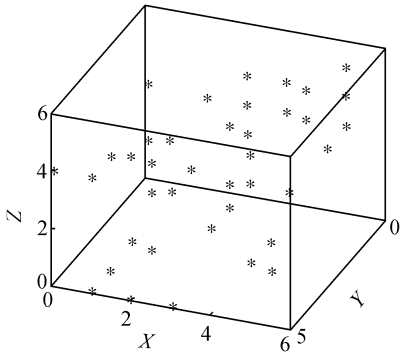

图 1　仿真评分数据在三维上的分布

Fig. 1　Simulation rating data distributed in three-dimensional space

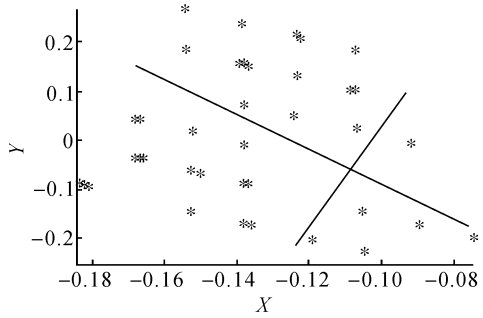

图 2　评分数据按 PCA 在二维上的投影

Fig. 2　Rating data projected in two-dimensional space according PCA

果距离相近，那么在低维的潜在兴趣空间距离也应该相近”的假设出发，容易找到有 Laplacian Eigenmap[16] 的局部不变性理论与之对应[17]，即：如果 2 个数据点 x_i 、x_j 在原来样本空间的几何上是相近的，那么降维后的 $\boldsymbol{h}_i$ 、$\boldsymbol{h}_j$ ，即在新基 W 上的 2 个数据点，也应该是相近的. 显然，这个理论指导思想对先降维再计算用户相似度的 CF 算法是重要的. 本文只需利用样本数据点内在结构信息，即 intrinsic graph 为 PNMF 施加约束，就可以实施无监督的降维. 本文开展一组仿真实验，图 3 反映了以相邻数据点数为 3 进行的降维投影的结果，图 4 反映了以相邻数据点数为 5 进行的降维投影的结果，可见不同邻接数对聚类效果不同，本文在实验结果分析中继续对邻接数进行讨论.

受文献[15]的启发，结合 PNMF 支持增量算法的优点，本文提出结构投影非负矩阵分解(SPNMF)模型，即对式(2)施加下列正则化约束：

$$\mathrm{R}=\frac{1}{2}\sum_{i,j=1}^{N}\|\boldsymbol{h}_i-\boldsymbol{h}_j\|^2V_{i,j}=\frac{1}{2}\sum_{i,j=1}^{N}\|\boldsymbol{P}\boldsymbol{x}_i-\boldsymbol{P}\boldsymbol{x}_j\|^2V_{i,j}=$$
$$\frac{1}{2}\sum_{i,j=1}^{N}(\|\boldsymbol{P}\boldsymbol{x}_i\|^2+\|\boldsymbol{P}\boldsymbol{x}_j\|^2-2\|\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}_j\|)V_{ij}=$$

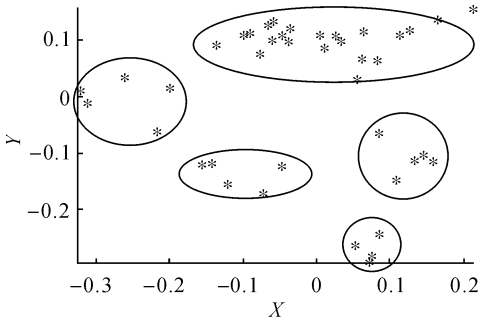

图 3　数据按 Laplacian 图以邻居数为 3 的投影

Fig. 3　Data projected in 2d according Laplacian graph when neighbours equal 3

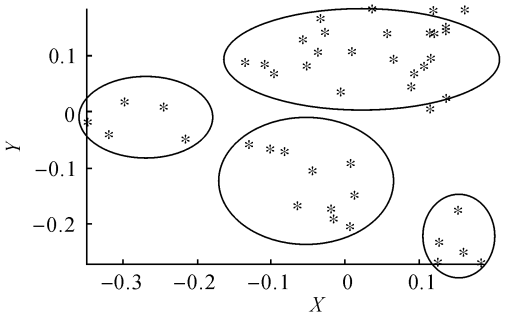

图 4　数据按 Laplacian 图以邻居数为 5 的投影

Fig. 4　Data projected in 2d according Laplacian graph when neighbors equal 5

$$\sum_{i=1}^{N}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}_iD_{ii}-\sum_{i,j=1}^{N}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x}_jV_{ij}=$$
$$\mathrm{Tr}(\boldsymbol{P}\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})-\mathrm{Tr}(\boldsymbol{P}\boldsymbol{X}\boldsymbol{V}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})=$$
$$\mathrm{Tr}(\boldsymbol{P}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}).\tag{4}$$

式中：Tr 为矩阵的迹函数，并利用了 $\mathrm{Tr}(\boldsymbol{A})=\mathrm{Tr}(\boldsymbol{A}^{\mathrm{T}})$ 以及 $\boldsymbol{V}$ 是对称矩阵的性质，$\boldsymbol{D}_{ii}$ 是对角矩阵，$D_{ii}=\sum_i V_{ij}$ ；同时，$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{V}$. Cai 等[18]进一步提出 3 种加权邻接矩阵来定义相近数据点：0－1 加权、热核加权、点积加权. 考虑到 2 个向量余弦相似性在向量归一化条件下是点积加权形式，选择两点之间余弦相似性的邻接矩阵来保存 2 个数据点 x_i 、x_j 的相近程度，即

$$V_{ij}=\frac{\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j}{\|\boldsymbol{x}_i\|\|\boldsymbol{x}_j\|}.\tag{5}$$

综上所述，SPNMF 模型的目标函数如下所示：

$$\left.\begin{aligned}O=&\min[\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{P}\boldsymbol{X}\|^2+\lambda\mathrm{Tr}(\boldsymbol{P}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})]=\\&\min[\mathrm{Tr}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})-2\mathrm{Tr}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}})+\\&\mathrm{Tr}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{P})+\lambda\mathrm{Tr}(\boldsymbol{P}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})];\\\text{s. t. }&\boldsymbol{W}\geqslant 0,\boldsymbol{P}\geqslant 0.\end{aligned}\right\}\tag{6}$$

与文献[15]相比，Liu 等所提模型是本文所提模型的一般形式，其模型在人脸分类和人脸识别上

得到应用,本文首次将该模型应用到推荐问题.两者相比,本文所用模型无需使用评分向量类别信息(实际上,评分矩阵也没有这样的先验信息),并且因为不用计算数据点之间的 penalty graph 的 Laplacian 图结构,从而使得在计算复杂度上更简单.

1.2 模型求解

式(6)是非凸函数,因此难以找到全局最小值,本文采用 Seung 等[19]提出的乘法迭代求解式(6)的局部极小值.

将目标函数 O 分别对 $\boldsymbol{W}$ 和 $\boldsymbol{P}$ 求导,可得

$$\frac{\partial O}{\partial \boldsymbol{W}} = -2\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} + 2\boldsymbol{W}\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}, \tag{7}$$

$$\frac{\partial O}{\partial \boldsymbol{P}} = -2\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} + 2\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} + 2\lambda\boldsymbol{P}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\mathrm{T}}. \tag{8}$$

通过 KKT 条件下 $\partial O/\partial \boldsymbol{W}=0$ 和 $\partial O/\partial \boldsymbol{P}=0$,可由式(7)和(8)得到

$$-(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})_{ij}\boldsymbol{w}_{ij} + (\boldsymbol{W}\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})_{ij}\boldsymbol{w}_{ij} = 0, \tag{9}$$

$$-(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})_{ij}\boldsymbol{p}_{ij} + (\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})_{ij}\boldsymbol{p}_{ij} + \lambda(\boldsymbol{P}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\mathrm{T}})_{ij}\boldsymbol{p}_{ij} = 0. \tag{10}$$

进一步可以表示为

$$w_{ij}^{t+1} \leftarrow w_{ij}^{t}\frac{(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})_{ij}}{(\boldsymbol{W}\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})_{ij}}, \tag{11}$$

$$p_{ij}^{t+1} \leftarrow p_{ij}^{t}\frac{(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} + \lambda\boldsymbol{P}\boldsymbol{X}\boldsymbol{V}\boldsymbol{X}^{\mathrm{T}})_{ij}}{(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{P}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} + \lambda\boldsymbol{P}\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}})_{ij}}. \tag{12}$$

2 基于 SPNMF 的协同过滤算法

2.1 算法描述

为了满足互联网推荐系统对算法精度和速度的要求,本文提出的基于 SPNMF 的协同过滤算法(CF-SPNMF)分成离线矩阵分解计算和在线推荐计算两部分.

第一部分是离线训练模型的过程,目的是为了把用户在高维评分空间的聚类转换到低维兴趣空间的聚类,由此发现更有鲁棒性的基于用户相似性的推荐算法.推荐系统在后台对评分矩阵 $\boldsymbol{X}$ 进行 SPNMF 后,获得代表用户兴趣特征的基矩阵 $\boldsymbol{W}$ 以及在基矩阵 $\boldsymbol{W}$ 下的投影系数矩阵 $\boldsymbol{P}$.在该过程中,为了解决评分矩阵的缺失评分项不能用 0 分代替的问题,采用 Zhang 等[20]所提的下列公式预先填充评分矩阵的缺失项:

$$\hat{x}_{i,u} = \alpha\bar{x}_i + (1-\alpha)\bar{x}_u. \tag{13}$$

式中:$\bar{x}_i$ 为缺失项所在行的平均分(即第 i 个物品的平均分),$\bar{x}_u$ 为缺失项所在列的平均分(即第 u 个用户的平均分),凸组合中的 α 为 0~1.0 的常数.

第二部分是对目标用户在线计算基于用户相似性的预测评分项过程.目标用户可以是 $\boldsymbol{X}$ 评分矩阵中的老用户(即训练集样本),也可以是新加入用户(即测试集样本).算法在线部分利用式(3)对目标用户在投影变换矩阵 $\boldsymbol{P}$ 上投影,得到 h_t;然后与训练集在投影变换矩阵 $\boldsymbol{P}$ 上投影 $\boldsymbol{P}\boldsymbol{X}=[h_1,\cdots,h_n]$,进行余弦相似度计算,找出最相近的前 k 个用户;最后根据式(14),用前 K 个最相似用户的已知评分项共同计算目标用户的未知评分项:

$$x_{jm} = \bar{u}_j + \frac{\sum_{u_i}\mathrm{sim}(u_i,u_j)(u_{im}-\overline{u_i})}{\sum_{u_i}\mathrm{sim}(u_i,u_j)}. \tag{14}$$

式中:x_{jm} 为目标用户 j 对待评项目的预测分,$\overline{u_j}$ 为目标用户 j 对已评项目的平均分,$u_i \in \{u_j$ 的前 k 个最相似用户$\}$.

当大量新增用户产生后会影响推荐精度(详见 3.4.5 节),此时 CF-SPNMF 需要重新离线计算基矩阵 $\boldsymbol{W}$ 和投影矩阵 $\boldsymbol{P}$,但对在线计算时间的影响很小.综上所述,总结基于结构投影非负矩阵分解的协同过滤算法(CF-SPNMF)如下.

算法离线部分.

输入:评分矩阵 $\boldsymbol{X}$,低维特征空间维数 d,模型参数 λ 和高维空间中的用户的邻接数 p;

输出:特征空间的非负基矩阵 $\boldsymbol{W}$,基矩阵 $\boldsymbol{W}$ 下的非负投影变换矩阵 $\boldsymbol{P}$;

1)用式(14)计算预先填充的评分矩阵 $\boldsymbol{X}$,α 为 0~1.0的常数;

2)用式(5)计算邻接矩阵 $\boldsymbol{V}$,对角阵 $D_{ii}=\sum_i V_{ij}$;

3)随机产生初始矩阵 $\boldsymbol{W}$ 与 $\boldsymbol{P}$;

4)while $t \leqslant$ max_iter 或 (O_old - O_new) $\leqslant 10^{-6}\times$ O_new

a) $\boldsymbol{W}^{t+1} = \boldsymbol{W}^{t}.*(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{P}^{t})^{\mathrm{T}})./(\boldsymbol{W}^{t}\boldsymbol{P}^{t}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{P}^{t})^{\mathrm{T}})$;

b) $\boldsymbol{P}^{t+1} = \boldsymbol{P}^{t}.*[(\boldsymbol{W}^{t+1})^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} + \lambda\boldsymbol{P}^{t}\boldsymbol{X}\boldsymbol{V}\boldsymbol{X}^{\mathrm{T}}]./[(\boldsymbol{W}^{t+1})^{\mathrm{T}}\boldsymbol{W}^{t+1}\boldsymbol{P}^{t}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} + \lambda\boldsymbol{P}^{t}\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}}]$;

c) O_old = C; O_new = $\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{P}\boldsymbol{X}\|^2 + \lambda\mathrm{Tr}(\boldsymbol{P}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}})$;

5)end while

算法在线部分.

输入:测试集的预测用户 x_j,第 m 项物品,投影变换矩阵 $\boldsymbol{P}$,原评分矩阵 $\boldsymbol{X}$,Top−k

输出:预测分 x_{jm}

1)计算 $\boldsymbol{H}=\boldsymbol{P}\boldsymbol{X}$,$\boldsymbol{H}=[\boldsymbol{h}_1,\cdots,\boldsymbol{h}_n]$;

2)按式(14)补全目标用户评分向量 $\boldsymbol{x}_j$,然后按式(3)获得 $\boldsymbol{x}_j$ 在特征空间上的投影 $\boldsymbol{h}_{x_j}$;

3)for i = 1 to n

计算 $\boldsymbol{h}_{x_i}$ 与 $\boldsymbol{h}_j$ 的余弦相似度，

$$\mathrm{sim}(\boldsymbol{h}_{x_i},\boldsymbol{h}_j)=\frac{\boldsymbol{h}_{x_i}^{\mathrm{T}}\boldsymbol{h}_j}{\|\boldsymbol{h}_{x_i}^{\mathrm{T}}\|\|\boldsymbol{h}_j\|};$$

4) end for

5) 排序 Top－k 个最相似用户 $\{\boldsymbol{h}_i,\cdots,\boldsymbol{h}_k\}$；

6) 用式(15)计算预测分 x_{jm}.

2.2　算法时间复杂度分析

对 n 个用户 m 个物品的评分矩阵而言，传统基于用户相似性的 CF 算法时间复杂度为 $O(n^2)$，本文提出的 CF-SPNMF 在线算法的部分时间复杂度为 $O(dn)$，因为 $d\ll n$，CF-SPNMF 在线计算时间比传统算法快. 另外，CF-SPNMF 离线算法的时间复杂度与实验部分所提算法的时间复杂度分析，详见表 1.

表 1　不同算法的时间复杂度

Tab. 1　Computational complexity of different algorithm

PMF	WNMF	CFONMTF	CF-SPNMF
$O(m^2n)$	$O(mn)$	$O(m^2n+n^2m)$	$O(m^2n)$

3　实验结果与分析

3.1　数据集和度量标准

分别使用 Netflix 数据集和 MovieLens 数据集来验证算法. 实验中，训练集和测试集是按照用户(即样本点)来划分，即将 MovieLens 的 80%用户作为训练集，其余作为测试集. 同时，从 Netflix 中随机抽取 10 000 个用户和 5 000 部电影作为样本，其中 80%用户作为训练集，其余作为测试集，如表 2 所示. 表中，u 为用户数，t 为项目数，r 为评分数，s 为稀疏率. MovieLens 和 Netflix 测试集的每个用户评分记录都被分为已知部分和对比部分. 实验结果的验证方法都是用已知部分预测对比部分的评分值，然后与对比部分的真实值进行比较，从而度量评分精度.

表 2　数据集描述

Tab. 2　Description of data sets

数据集	u	t	r	s
MovieLens	6 040	3 952	1 000 209	95%
Netflix	10 000	5 000	2 050 082	96%

评价推荐系统质量的度量标准主要包括平均绝对误差 MAE 和均方根误差 RMSE，本文选用 MAE 作为度量标准. MAE 越小，预测精度越高. 设第 j 个目标用户的预测评分集合为 $\{x_{j1},x_{j2},\cdots,x_{jN}\}$，则 MAE 定义为

$$\mathrm{MAE}=\frac{\sum_{j,m}|x_{jm}-\hat{x}_{jm}|}{N}. \tag{16}$$

3.2　与其他算法的比较

分别选择目前主流的 PMF 算法[18]、加权非负矩阵分解(WNMF)算法[20]、正交非负矩阵三分解(CFONMTF)算法[21]与本文算法进行比较. 实验分别选择 2 个数据集的 20%、50%、80%作为训练集，低维空间维度 $d=[10,20,30]$，实验精度的对比如表 3 所示. 可见，当训练集大小占数据集的 20%，测试集占 80% 的时候，PMF 的预测精度优于 CF-SPNMF，而且 WNMF 在维度大小为 10 和 20 的情况下，预测精度优于 CF-SPNMF. 原因主要是因为训练集样本比测试集少很多，无法用少量样本邻接图结构表达大量未知样本的结构. 当训练集大小超过测试集大小时，CF-SPNMF 的预测精度优于另外 3 个算法的结果，并且随着训练集的扩大和测试集中用户评分项的增多，算法精度越来越高. 这种通过大数据学习，不断提升算法精度的特点非常适用于互联网推荐系统的环境.

表 3　与其他算法在 MAE 度量上的比较

Tab. 3　Comparison with other algorithm on MAE metric

数据集	算法	20%作为训练集			50%作为训练集			80%作为训练集		
		$d=10$	$d=20$	$d=30$	$d=10$	$d=20$	$d=30$	$d=10$	$d=20$	$d=30$
MovieLens	PMF	0.752	0.745	**0.697**	0.753	0.749	0.688	0.733	0.695	0.690
	WNMF	**0.741**	**0.740**	0.719	0.745	0.735	0.714	0.728	0.711	0.709
	CFONMTF	0.897	0.876	0.843	0.898	0.888	0.857	0.891	0.825	0.822
	CF-SPNMF	0.758	0.755	0.710	**0.728**	**0.695**	**0.680**	**0.704**	**0.665**	**0.651**
Netflix	PMF	**0.775**	0.771	**0.749**	0.757	0.742	0.732	0.724	0.715	0.699
	WNMF	0.792	0.784	0.762	0.799	0.751	0.744	0.770	0.723	0.715
	CFONMTF	0.909	0.925	0.872	0.898	0.887	0.873	0.861	0.865	0.850
	CF-SPNMF	0.798	**0.770**	0.755	**0.785**	**0.731**	**0.727**	**0.745**	**0.706**	**0.681**

3.3　参数选择

由于篇幅所限，本文不再介绍不同训练集、测试集下参数对算法影响的细节，选择 Netflix 数据集作为实验测试环境，重点关注 SPNMF 模型的维度 d、

系数和邻居数 p、预先填充评分矩阵的参数以及新增用户数 n 对算法精度的影响. 设定 $\lambda=0.5$, $d=30$, $p=7$, $\alpha=0.2$, $n=1\,000$ 为基本参数,通过调整其中一项参数值进行实验.

3.4.1 特征向量维度 d 对算法的影响 实验中,分别设定基矩阵维度 $d=5$、10、20、25、30、40. 图 5 反映了维度 d 对算法精度的影响:随着 d 的增大,预测精度都有一定的提高. 需要指出的是,d 越大越耗时.

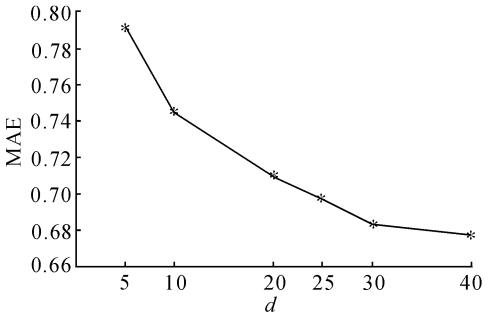

图 5 维度 d 的影响

Fig. 5 Influence of dimension d on algorithm

3.3.2 邻居数 p 对算法的影响 实验中,分别设定 $p=3$、5、7、9、11、13、15. 如图 6 所示,高维数据点的不同邻接数量在降维过程中,对 CF-SPNMF 算法的精度有直接影响. 如果 p 高于原始评分数据点的真实邻居数,那么会把原本不相邻点拉到一起;如果 p 小于原始评分数据点的真实邻居数,那么会产生多余的聚类结构. 图 6 表明,训练集以 7 或 13 为邻居数最能够反映出原高维数据点分布结构的特征.

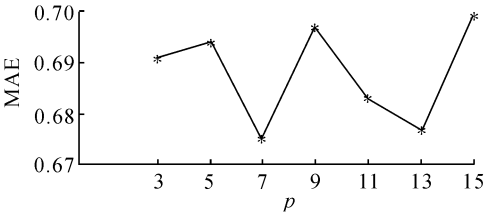

图 6 邻居数 p 的影响

Fig. 6 Influence of neighbor p on algorithm

3.3.3 模型系数对算法的影响 可以衡量正则化项对算法的影响程度,λ 越大,表明算法的正则化项对算法精度的影响越小. 实验中,$\lambda=0.1$、0.4、0.8、1、1.5、5. 图 7 表明,当 λ 较小时,对 MAE 结果的影响较大;随着 λ 的增大,算法精度逐步下降. 当 $\lambda=0.8$ 时,预测精度最佳.

3.3.4 预先填充评分矩阵的 α 参数对算法影响 在实验中,直接运用已知用户和物品的平均分或其凸组合对评分矩阵进行预填充,以避免矩阵分解时用 0 分代替对未评分数据的问题. 这种简单的处理

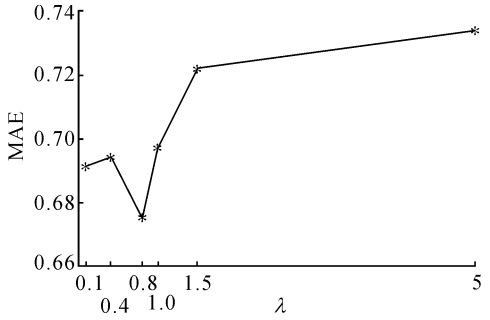

图 7 系数 λ 的影响

Fig. 7 Influence of coefficient λ

方式会遇到实际问题,即 Netflix 数据集上反映出用户和物品的平均分非常趋同:好的电影平均分一般为 3.5~4.5,而对电影评分多的用户的平均分一般约为 3.5. 这对算法是不利的,因为目标用户的兴趣被淹没在大众兴趣中,这和用 0 分代替评分缺项效果差不多. 实验选择 $\alpha=0.2$、0.4、0.6、0.8、1.0. 图 8 的结果表明,α 对算法的影响不大,且抑制物品权重($\alpha=0.2$)对拉开预填项分值有帮助,对提高 MAE 结果也有帮助.

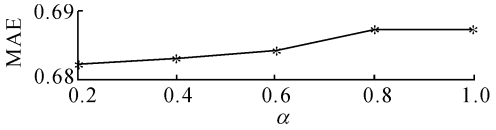

图 8 参数 α 的影响

Fig. 8 Influence of parameter α

3.3.5 新增用户数对算法精度影响 在实验中,分别选择整个 Netflix 数据集的 2%、5%、10%、15%、20%、25%作为新增用户数,在不重新计算矩阵分解的情况下,直接把新增用户在线投影到低维 P 空间并与训练集用户作相似性匹配,由此产生预测精度. 图 9 表明,随着新增用户数 p 的增大,预测精度呈线性降低,由此可见算法对增量用户数据的可扩展性较好.

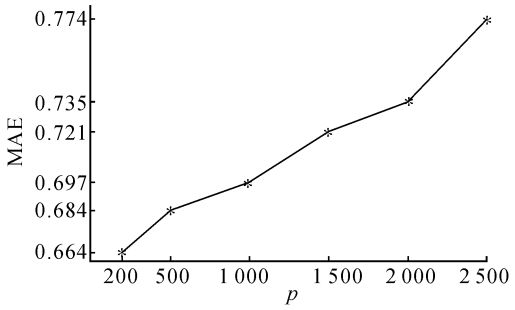

图 9 算法的可扩展性

Fig. 9 Scalability of algorithm

4 结　语

本文利用 Laplacian Eigenmap 的局部不变性理论对投影非负矩阵分解进行正则化项约束，并推导出迭代学习算法，能够有效地学习评分矩阵中的代表用户潜在兴趣因子的基矩阵. 在真实数据集上的实验表明，用该模型构造的两阶段协同过滤推荐算法与传统的推荐算法相比，有较大的精度提高. 该算法对新加入的用户具有很好的扩展性，因此非常适用于互联网的推荐系统. 该算法遇到的预填充用户和物品平均分的趋同问题，将在今后的工作中进一步研究.

参考文献(References)：

[1]LU L, MEDO M, YEUNG C H, et al. Recommender systems [J]. **Physics Reports**, 2012, 519(1): 1-49.

[2]SU X, KHOSHGOFTAAR T M. A survey of collaborative filtering techniques [J]. **Advances in Artificial Intelligence**, 2009, 2009(10): 4-24.

[3]LINDEN G, SMITH B, YORK J. Amazon. com recommendations: item-to-item collaborative filtering [J]. **Internet Computing, IEEE**, 2003, 7(1): 76-80.

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

[5]KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems [J]. **Computer**, 2009, 42(8): 30-37.

[6]PATEREK A. Improving regularized singular value decomposition for collaborative filtering[C]// **Proceedings of KDD Cup and Workshop.** California: ACM, 2007: 5-8.

[7]ZHANG S, WANG W, FORD J, et al. Using singular value decomposition approximation for collaborative filtering [C]// **7th IEEE International Conference on E-Commerce Technology.** Ankara: IEEE, 2005: 257-264.

[8]SREBRO N, JAAKKOLA T. Weighted low-rank approximations [C]// **ICML.** Washington: DBML, 2003: 720-727.

[9]MNIH A, SALAKHUTDINOV R. Probabilistic matrix factorization [C]// **Advances in Neural Information Processing Systems.** Vancouver: MIT, 2007: 1257-1264.

[10]LEE D D, SEUNG H S. Learning the parts of objects by non-negative matrix factorization [J]. **Nature**, 1999, 401(6755): 788-791.

[11]HOYER P O. Non-negative sparse coding [C]// **Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing.** London: IEEE, 2002: 557-565.

[12]**胡俐蕊，吴建国，汪磊. 线性投影非负矩阵分解方法及应用**[J]. **计算机科学**, 2013, 40(10): 269-273.
HU Li-rui, WU Jian-guo, WANG Lei. Application and method for linear projective non-negative matrix factorization [J]. **Computer Science**, 2013, 40(10): 269-273.

[13]YUAN Z, YANG Z, OJA E. Projective nonnegative matrix factorization: sparseness, orthogonality, and clustering [J]. **Neural Processing Letters**, 2009, 2009(1): 33-47.

[14]YANG J, YANG S, FU Y, et al. Non-negative graph embedding [C]// **IEEE Conference on Computer Vision and Pattern Recognition.** Hongkong: IEEE, 2008: 1-8.

[15]LIU X, YAN S, JIN H. Projective nonnegative graph embedding [J]. **IEEE Transactions on Image Processing**, 2010, 19(5): 1126-1137.

[16]BELKIN M, NIYOGI P. Laplacian eigenmaps and spectral techniques for embedding and clustering [C]// **NIPS.** Vancouver: MIT, 2001: 585-591.

[17]NIYOGI X. Locality preserving projections [C]// **Neural Information Processing Systems.** Los Angeles: MIT, 2004: 153-161.

[18]CAI D, HE X, HAN J, et al. Graph regularized nonnegative matrix factorization for data representation [J]. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2011, 33(8): 1548-1560.

[19]SEUNG D, LEE L. Algorithms for non-negative matrix factorization [J]. **Advances in Neural Information Processing Systems**, 2001, 2001(3): 556-562.

[20]ZHANG Z, ZHAO K, ZHA H. Inducible regularization for low-rank matrix factorizations for collaborative filtering [J]. **Neurocomputing**, 2012, 97(1): 52-62.

[21]HU Li-rui, WU Jian-guo, WANG Lei. Application and method for linear projective non-negative matrix factorization [J]. **Computer Science**, 2013, 40(10): 269-273.