ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.5, May 2010, pp.1042−1054
doi: 10.3724/SP.J.1001.2010.03758

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

两阶段联合聚类协同过滤算法*

吴 湖[1,3+], 王永吉[1,2], 王 哲[2,3], 王秀利[4], 杜栓柱[1]

[1](中国科学院 软件研究所 互联网软件技术实验室,北京 100190)

[2](中国科学院 软件研究所 计算机科学国家重点实验室,北京 100190)

[3](中国科学院 研究生院,北京 100049)

[4](中央财经大学,北京 100081)

Two-Phase Collaborative Filtering Algorithm Based on Co-Clustering

WU Hu[1,3+], WANG Yong-Ji[1,2], WANG Zhe[2,3], WANG Xiu-Li[4], DU Shuan-Zhu[1]

[1](Laboratory for Internet Software Technologies, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](State Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[3](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

[4](Central University of Finance and Economics, Beijing 100081, China)

+ Corresponding author: E-mail: wuhu@itechs.iscas.ac.cn

Wu H, Wang YJ, Wang Z, Wang XL, Du SZ. Two-Phase collaborative filtering algorithm based on co-clustering. *Journal of Software*, 2010,21(5):1042−1054. http://www.jos.org.cn/1000-9825/3758.htm

Abstract: This paper proposes a two-phase rating predicting framework that fuses co-clustering and non-negative matrix factorization method. First, it uses a novel co-clustering method (BlockClust) to divide the raw rating matrix into clusters much smaller than the original matrix. Then it employs weighted non-negative matrix factorization algorithm to predict the unknown ratings. In virtue of co-clustering preprocessing, this method achieves a higher predicting accuracy and efficiency on these low-dimensional and homogeneous sub-matrices. Moreover, it proposes three update schemes for the corresponding update scenarios in recommender systems. Finally, the proposed method is implemented together with seven types of related CF (collaborative filtering) methods. The comparisons show the efficiency of the proposed method and its potential in large real-time recommender systems.

Key words: collaborative filtering; recommender system; co-clustering; NMF (non-negative matrix factorization)

摘 要: 提出一种两阶段评分预测方法.该方法基于一种新的联合聚类算法(BlockClust)和加权非负矩阵分解算法.首先对原始矩阵中的评分模式进行用户和物品两个维度的联合聚类,然后在这些类别的内部通过加权非负矩阵分解方法进行未知评分预测.这种方法的优势在于,首阶段聚类后的矩阵规模远远小于原始评分矩阵,并且同一类别内部的评分具有相似的模式,这样,在大幅度降低预测阶段计算量的同时又提高了非负矩阵分解算法在面对稀疏矩阵预测上的准确度.进一步给出了推荐系统的 3 种更新模式下如何高效更新预测模型的增量学习方法.在

* Supported by the National Natural Science Foundation of China under Grant Nos.60673022, 60673121 (国家自然科学基金); the State Key Laboratory of Computer Science Funding for Innovative Research of China under Grant No.CSZZ0808 (计算机科学重点实验室自主研究课题)

Received 2009-05-07; Revised 2009-07-06; Accepted 2009-10-19

MovieLens数据集上比较了新算法及其他7种相关方法的性能,从而验证了该方法的有效性及其在大型实时推荐系统中的应用价值.

关键词: 协同过滤;推荐系统;联合聚类;非负矩阵分解

中图法分类号: TP311 **文献标识码**: A

推荐系统(recommender system)的目标是根据网络用户的个性化需求将最符合用户兴趣的信息挑选出来并且推荐给用户,随着互联网上信息的增长和用户个性化需求的提高,推荐系统的应用日益广泛,成为电子商务、社会网络、视频/音乐点播等主流 Web 2.0 服务的核心技术[1].协同过滤(collaborative filtering)是推荐系统所采用的最为重要的技术之一,其原理是根据相似用户的兴趣来推荐当前用户没有看过但是很可能会感兴趣的信息,所基于的假设是,如果两个用户兴趣类似,那么很有可能当前用户会喜欢另一个用户所喜欢的内容.协同过滤算法的优势在于不受被推荐的物品的具体内容的限制、与社会网络的紧密结合以及推荐的准确性.

协同过滤或者是其他类型的推荐系统普遍面临的挑战是如何面对互联网环境下的海量数据做出准确的推荐,其难点有 3 个[2]:(1) 数据量巨大,需要推荐算法能够在尽可能短的时间内作出响应;(2) 数据的稀疏性,这看起来与数据量巨大是矛盾的,但是相对于系统中为数众多的用户和待推荐的物品,我们能够利用的表示用户兴趣的信息(一般是用户对自己感兴趣信息的评分)实际上是非常稀疏甚至有限的;(3) 数据的动态性,推荐系统中不断有新的数据加入,而且用户的兴趣和关注点也在不断地改变,用户在使用的过程中还在不断增加新的训练数据,要求推荐算法能够快速、准确地进行更新.

对于前两个困难,许多研究者提出了很多利用聚类或者说数据降维的方法来解决,这其中包括基于概率的模型,如隐含主题分析(probabilistic latent semantic analysis,简称 PLSA)[3]、隐含 Dirichlet 分析(latent Dirichlet allocation,简称 LDA)[4]等;基于矩阵分解的模型,如奇异值分解(singular value decomposition,简称 SVD)[5]、非负矩阵分解(non-negative matrix factorization,简称 NMF)[6]等.还有能够同时在多个维度进行聚类的联合聚类(co-clustering)方法[2,7−9],也逐渐在推荐系统中得到了应用.这些方法能够有效地降低训练数据的维度,有些还能有效地降低数据的稀疏性,相对于传统的直接计算用户或者物品相似度的方法,在提高准确性的同时,减小了推荐,也就是在线(online)的计算量.然而,以上这些方法都存在离线(offline)计算量较大、模型更新较为困难的缺陷,所以不能很好地解决前述第 3 个问题.面对不断更新的信息和用户兴趣,本文提出一种两阶段降维预测方法,即首先利用联合聚类的方法将用户和待推荐的物品聚成一些小类,然后再使用 NMF 方法来进一步填充这些小类中的位置元素,达到整体评分预测的目的.

本文的贡献包括如下3个方面:(1) 提出了一种两阶段降维的增量式推荐方法,使得利用新的训练数据能够很快地更新学习模型;(2) 我们使用一种联合软聚类(soft co-clustering)方法来对原始数据进行首次降维,可以非常灵活地按照评分特征将评分聚成子类,特别适合评分预测问题;(3) 提出了一种改进的 NMF 初始化方案来进行二次降维,有效地弥补了 NMF 方法在进行评分数据预测时的缺陷.

本文第 1 节介绍相关的研究——联合聚类和非负矩阵分解.第 2 节给出我们的算法描述.第 3 节给出在有新的评分加入训练数据集时的增量式更新方法.第 4 节是我们的方法在 MovieLens 电影评分数据集上与其他相关方法进行对比的实验结果.最后,我们总结全文并指出未来的工作.

1 相关研究

在此我们首先给出本文中所使用的记号及其含义.

1.1 协同过滤

推荐系统中有 m 个用户集合 U 和 n 个物品集合 V,U 中的用户对于 V 中的物品有选择性地进行了评分,分值为$[1,\mathrm{Max}_{\mathrm{rating}}]$区间内的整数,分值的增加表示感兴趣程度依次递增,这样就形成了一个评分矩阵 R,两个维度分别代表用户集合和物品集合,其中行向量是用户所评价过的分值,列向量是物品被评价的分值.协同过滤的目

标就是通过 R 矩阵已知的评分去预测其中未知项(记为 0)的评分,因此,学习的目标就转化为对于一个非负稀疏矩阵 R 中的未知项进行预测和填充.

Table 1 Notations used in this paper

表 1 本文使用的记号列表

Notation	Meaning	Notation	Meaning
R	Rating matrix	$C(u),C(v)$	The cluster set that user u belongs to and the item v belongs to
U,u	User set and current user	$U(k),V(k)$	Users and items that cluster k contains
V,v	Item set and current item	W,H	Non-Negative matrix factorization results
m,n	Number of users and items	(u,v,r)	A rating record
K,k	Cluster set and current cluster	$V(u)$	The item set that the user u has rated before
$p(k\mid u,v,r)$, $p(k\mid u)$, $p(k\mid v)$, $p(r\mid k)$	Probability of cluster given user, item and rating, probability of cluster given user, probability of cluster given item and probability of rating given cluster	$U(v)$	The user set which have rated item v before

早期的协同过滤算法称为基于内存(memory-based)的方法,原理是计算用户或物品的相似度,再用较为相似的用户或者物品的已知评分去加权预测未知评分项.由于存在着过多的未知评分元素,导致相似度的计算很难代表两项间的真实距离,因此,基于模型(model-based)的方法被提出并成为研究的热点[2].此类方法将原始评分矩阵拟合为一个约简的、稀疏性降低的模型(概率分布或者低维矩阵),然后通过这个模型对未知评分进行预测.基于模型的方法的本质在于降低数据的维度.它们是基于这样的观察:通常决定用户对物品的评分的因素可以归结为维度相对较少的一些隐含的要素(如电影类型、商品类型等),同一类型内的用户对同一类型内的商品往往有相似的评分.基于模型的方法可以分为聚类的方法、概率的方法和矩阵分解的方法等大类,更详细的算法描述参见文献[2,5].

以上提到的这些方法普遍面临的一个棘手问题就是当用户的评分增加或改变时,如何实时地更新推荐内容.由于推荐系统内的用户的兴趣经常发生改变,新的评分也不断地加入原有评分矩阵,如何提高推荐的实效性和灵活性是一个重要而很有挑战性的问题.通常的解决方法是通过增量式学习的方法,也就是考虑最新的学习样本更新现有模型而不是重新学习一个全新的模型.目前,协同过滤算法多数缺乏增量学习的能力.本文通过两阶段学习的方法提出了一种有效的解决方案.

1.2 联合聚类

聚类(clustering)是将具有类似属性的内容聚集在一起的无监督机器学习方法.如何处理海量同时又很稀疏的训练数据是协同过滤算法所面临的核心问题,因此将原始训练数据通过聚类手段划分成相似度较高、数据规模较小的子类别成为一种非常常见且有效的方法.对于协同过滤问题,采用聚类算法理由有两点:(1) 减少复杂操作所需要处理的数据规模;(2) 减少评分稀疏性.自从协同过滤的概念提出以来,不断有研究者使用聚类方法来寻找相似的用户或者相似的物品进而做出推荐.联合聚类又称为二部聚类(bi-clustering),是聚类方法的一种,最初用于基因表达(gene expression)[7]、文本分析[10]等领域,可以同时对基因和所处表达环境或者文本以及单词进行聚类.一般情况下,对于使用矩阵方式表达的训练数据,当行和列同时具有相关性时,应当考虑使用联合聚类,因为无论从哪一个维度进行单独聚类时,都会忽略另一维的相关信息.这一思想也体现在协同过滤评分预测方法中[11].

联合聚类的基本原理是通过行聚类和列聚类两个步骤进行循环迭代直至收敛.Dhillon 等人[12]提出了一种以 Kullback-Leibler(KL)**距离最小为标准的联合聚类方法;Banerjee 等人[13]提出了一种同时考虑类别内部均值

** K-L 距离定义为 $D_{KL}(P\|Q):=\sum_i P(i)\log\frac{P(i)}{Q(i)}$.

和行与列向量全局均值的联合聚类方法,并且给出了该方法在 MovieLens 数据集上进行评分预测的结果;Agarwal 等人[14]提出了利用扩展线性模型(generalized linear model)来平滑误差函数的聚类方法,并提供了一种基于 Expectation Maximization(EM)的模型拟合方法.在此基础上,文献[15]提出了如何设定聚类的类别个数的参考标准.

以上的方法均属于硬(hard)聚类,也就是说,某一行或某一列只能属于一个类别.与聚类中的软(soft)聚类相类似,Shafiei 等人提出了软联合聚类,放宽了对类别归属的限制,并参照 LDA 模型[4]给出了使用 Gibbs Sampling 进行模型学习的方法[16].软聚类更符合协同过滤问题的需要,因为无论是用户还是物品,都不太可能恰好只属于某一个类别[9].

此外,还有一些研究将联合聚类直接应用于评分预测.George 等人[17]提出一种直接使用联合聚类进行评分预测的方法:COCLUST.COCLUST 使用用户评分均值、物品评分均值、用户所属类别评分均值、物品所属类别评分均值和联合类别均值这 5 个偏移量对原始评分进行矫正并进行联合聚类,同时,提供了并行计算和增量学习两种加速方案.Chen 等人[8]使用一种称为正交非负矩阵分解的方法来预测评分,这是一种在 NMF 分解的基础上增加了正交约束的优化方法,同时也可以看成是一种联合软聚类方法.类似的方法还有 Long 等人提出的三元组分解方法[18].这些方法都可以看成是 NMF 的扩展方法.

1.3 非负矩阵分解及加权非负矩阵分解(weighted NMF)

非负矩阵分解可以将任意一个非负矩阵分解成两个非负矩阵的乘积形式,通过限制分解后矩阵的维度可以达到降低数据维度的目标,同时对于矩阵中的未知元素,也具有拟合预测的能力.为了解决协同过滤问题,处理评分矩阵 R,对其进行非负矩阵分解等价于下面的优化问题:

$$\min \| R_{m\times n} - W_{m\times k} H_{k\times n} \|_F^2 , \text{ s.t. } W \geq 0, H \geq 0, k < \min(m,n).$$ ***

我们注意到 R 矩阵是一个极为稀疏的矩阵,这意味着,如果不对其中的零元(也即未知评分项)单独进行处理,学习得到的新的低维矩阵就尽可能地在这些位置填充零元,这与我们预测这些未知评分项的目标是相悖的.所以在计算误差时需要剔除这些零元,仅考虑 R 矩阵中的非零元,这就是加权 NMF 的基本原理.方法是首先得到一个示性矩阵 $Indicator$,定义为 $\begin{cases} indicator_{ij} = 0, & \text{if } R_{ij} = 0 \\ indicator_{ij} = 1, & \text{if } R_{ij} \neq 0 \end{cases}$,然后最小化下面的损失函数:

$$\min \| Indicator_{m\times n} \otimes (R_{m\times n} - W_{m\times k} H_{k\times n}) \|_F^2 , \text{ s.t. } W \geq 0, H \geq 0, k < \min(m,n),$$

其中,⊗代表矩阵按元素乘(element-wise multiplication).相应的迭代算法是[6]:

$$W_{ij}^{(t+1)} = W_{ij}^{(t)} \frac{((Indicator \otimes R)H^T)_{ij}}{((Indicator \otimes (WH))H^T)_{ij}} \tag{1}$$

$$H_{ij}^{(t+1)} = H_{ij}^{(t)} \frac{(W^T(Indicator \otimes R))_{ij}}{(W^T(Indicator \otimes (WH)))_{ij}} \tag{2}$$

NMF 能够有效降低数据维度,并且得到的分解结果具有直观物理意义(W 矩阵代表用户的兴趣维度,H 矩阵代表物品的特征维度);然而,NMF 也存在着明显的缺陷:(1) 计算复杂度高,不仅每次迭代的复杂度是 $O(\| U \| \times \| K \| + \| V \| \times \| K \|)$,而且收敛缓慢,实验中需要约 200 次迭代才可以得到令人较为满意的结果;(2) NMF 的分解结果存在局部最小化的问题,并且与源矩阵的行和列的顺序有密切关系;(3) 学习拟合算法是针对整个评分矩阵进行优化,因此考虑了过多的非相似用户和非相似物品的冗余评分信息,导致结果的不精确.

为了降低计算的复杂度,并且尽可能多地使用相似的用户和物品的评分同时排除非相关评分信息的干扰,我们无疑需要先对评分矩阵进行预处理.其方法就是通过聚类缩小 NMF 所需要处理的数据规模和提高类别内部评分数据质量及密度.为此,我们提出了基于联合聚类的两阶段评分预测方法.

*** 矩阵 A 的 Frobenius 矩定义为 $\| A \|_F := \sqrt{\sum_{1\leq i,j\leq(m,n)} a_{ij}^2}$.

2 两阶段聚类预测算法

如图 1 所示,我们的算法分为两个步骤,首先对原始评分矩阵进行联合聚类,之后在得到的类别上分别利用我们改进过的 NMF 方法来预测类别中的未知评分.采取两阶段策略的理由主要有两点:首先,评分预测方法具有局部特征,也就是说,当评分矩阵行或者列的顺序调整之后,预测的结果会有所区别,当相似用户和相似物品位于同一子集时,通过对这个子集进行评分预测,实验结果表明其效果比考虑整体评分矩阵要好;其次,通过首先进行联合聚类,可以将用户对物品的评分的影响局限于它们所属的子类中,当新的评分到来时,我们只需要调整这个类别内部的其他评分预测结果,从而快速、准确地反映了最新的数据特征.

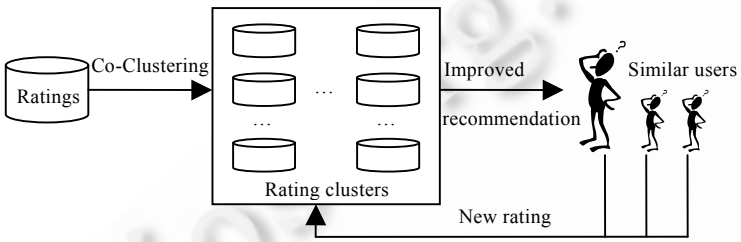


Fig.1 Workflow of the proposed algorithm

图 1 算法流程

2.1 联合聚类

针对协同过滤的评分矩阵进行联合聚类,已有的算法存在两个问题:

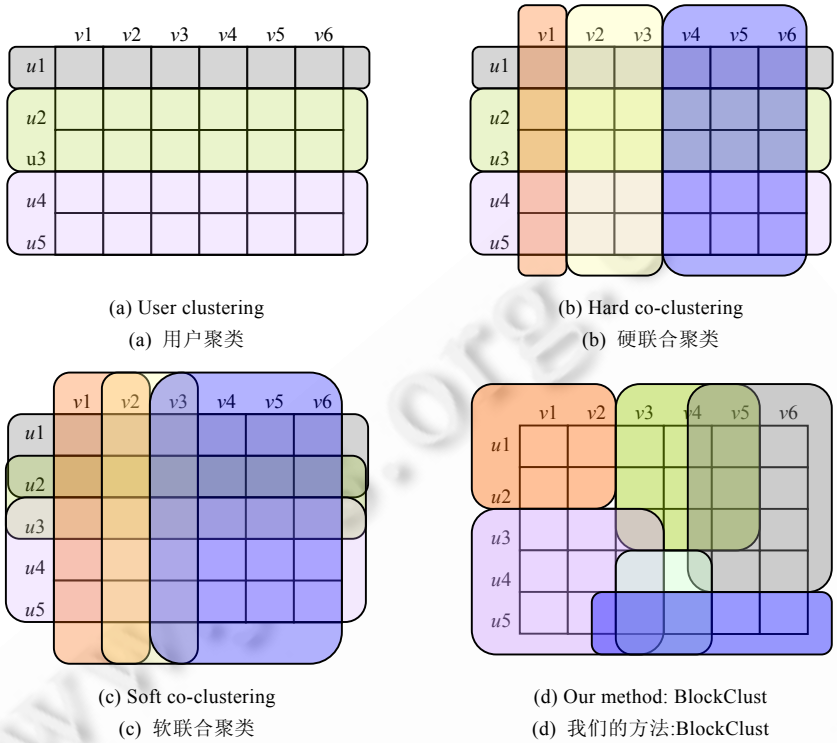
- 聚类方法本质上都是缩小类别内部的数据间的差异,如何计算评分间的差异,协同过滤问题与文档聚类等其他数据挖掘领域中的所谓同现(co-occurrence)数据有很大的区别,首先是距离计算时应当仅考虑已有评分项,其次是评分间相似度计算标准不同,比如 1 分与 5 分有很大的区别,而在文本分析领域则认为出现 1 次和 5 次并没有本质上的不同.
- 无论是现有的硬联合聚类还是软联合聚类算法都不能很好地表征推荐系统中用户兴趣的类别.我们注意到,推荐系统中用户的类别和物品的类别是有紧密联系的,所以在我们提出的类别是用户和物品的共享的类别,也即同时将用户和物品划分到这些类别中,当且仅当该用户和该物品均属于该类别时,我们将其评分作为该类别中的评分.

综合以上两点考虑,我们提出了以评分模式为标准的联合聚类算法,称为 BlockClust.图 2 以 5 个用户、6 个物品的评分矩阵为例,对比了我们的聚类方法与其他 3 种聚类模式的区别.图 2(a)为普通用户维度的聚类示意图;图 2(b)和图 2(c)均为用户和物品两个维度的联合聚类,区别在于,图 2(b)中每个用户或物品只能属于一个类别,而图 2(c)中的可以属于多个类别,因此它们分别属于硬聚类和软聚类;图 2(d)为 BlockClust 的示意图,可以看出,其聚类模式比软聚类更为灵活,每个类别可以包含若干个用户和若干个物品的部分评分,而且每个评分可能属于多个类别.具体来说,BlockClust 扫描评分矩阵,计算每个评分属于某类别的概率 $p(k|u,v,r)$,原则是通过综合考虑该评分所涉及的用户和物品属于这个类别的概率以及该分值出现于这个类别的概率,然后通过累计用户所有的评分和物品所有的分值,我们可以分别得到相应的用户和物品属于某类别的概率 $p(k|u)$以及 $p(k|v)$,同时可以得到的还有某类别中出现某个评分的概率 $p(r|k)$.该过程迭代直至收敛.

$$p(k\,|\,u,v,r)=\frac{[p(k\,|\,u)+\alpha]\times[p(k\,|\,v)+\beta]\times[p(r\,|\,k)+\vartheta]}{\sum_{k'}[p(k'\,|\,u)+\alpha]\times[p(k'\,|\,v)+\beta]\times[p(r\,|\,k')+\vartheta]} \tag{3}$$

其中,α,β,θ是为了防止出现 0 分母而设置的超参数,在我们的实验中均一化为 0.000 000 01.

$$p(k\,|\,u)=\frac{\sum_{v=V(u)}p(k\,|\,u,v,r)}{\sum_{z'}\sum_{v=V(u)}p(z'\,|\,u,v,r)} \tag{4}$$



(a) User clustering (a) 用户聚类

(b) Hard co-clustering (b) 硬联合聚类

(c) Soft co-clustering (c) 软联合聚类

(d) Our method: BlockClust (d) 我们的方法:BlockClust

Fig.2 Co-Clustering illustration

图 2 联合聚类示意图

$$p(k\mid v)=\frac{\sum_{u=U(v)}p(k\mid u,v,r)}{\sum_{z'}\sum_{u=U(v)}p(z'\mid u,v,r)} \tag{5}$$

$$p_{discrete}(r\mid k)=\frac{\sum p(k\mid u,v,r)}{\sum_{r'}\sum\sum p(k\mid u,v,r')} \tag{6}$$

$$p_{continuous}(r\mid k)=\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left[-\frac{(r-\mu_k)^2}{2{\sigma_k}^2}\right] \tag{7}$$

其中, $\mu_k=\sum_r p_{discrete}(r\mid k)\times r,{\sigma_k}^2=\sum_r(r-\mu_k)^2\times p_{discrete}(r\mid k)$.

2.2 预 测

在预测阶段,我们可以选用多种矩阵填充方法进行最后的评分预测,比如简单的基于近邻的预测方法、基于矩阵分解的方法 NMF 和基于图模型的方法 PLSA[19].综合考虑时间复杂度和准确性,我们采用加权非负矩阵分解(WNMF)的方法来进行稀疏矩阵的评分预测.对于聚类后的子评分矩阵,WNMF 方法可以在很短的时间内收敛到局部最小[20],并且实验结果表明评分预测的效果优异.

实验中,我们发现 NMF 的最终分解结果与其初始值密切相关,这也印证了文献[21]中关于初始值的讨论.因此,挑选合适的初始值对于最终的预测结果至关重要.

当选用最为常用的随机化初始方法时,我们观察到这样的现象:用户 *A* 有少量比较低(高)的评分,而用户 *B* 则所有的评分都比较高(低),使用 WNMF 预测的结果往往是 *A* 的其他评分高于 *B* 的其他评分;对于物品也有类似的现象.这样的结果主要是因为随机初始化给予所有用户(物品)的兴趣向量从向量矩的角度来说是相当的,

因此,当用户 i 对于某物品 j 有较低评分时,学习算法倾向于正交化相应的用户向量 W_i 和物品向量 H_j^T,这使得用户 i 对其他物品 k 的预测评分 $W_i \cdot H_k^T (k \neq j)$ 易于变大(向量夹角一般较小),造成了上述现象.因此我们认为,对于 W 和 H 的初始化应考虑用户和物品已有的平均评分,也就是说,对于已有平均评分高的,应给予较高的初始化值;反之,则给予较低的初始化值.改进后的算法称为 Improved NMF(INMF).

2.3 算法及复杂度分析

为了预测评分矩阵中的未知项,我们的算法采取两阶段策略,具体算法如下:

算法 1. 两阶段评分预测算法.

第 1 阶段:联合聚类 BlockClust.

输入:评分矩阵、类别数$\|K\|$.

输出:每个类别所对应的用户和物品集合.

Step 1. 随机初始化 $p(k|u,v,r)$,使得 $\sum_k p(k \mid u,v,r) = 1$.

Step 2. 根据公式(4)重新计算 $p(k|u)$.

Step 3. 根据公式(5)重新计算 $p(k|v)$.

Step 4.1. 根据公式(6)计算分值概率 $p(r|k)$.

Step 4.2. 根据公式(7)对 $p(r|k)$进行高斯平滑.

Step 5. 根据公式(3)重新计算 $p(k|u,v,r)$,并选取概率最大的 k 作为该评分的类别.

Step 6. 跳转 Step 2,直至收敛.

第 2 阶段:非负矩阵分解评分预测

输入:每个类别所对应的用户和物品集合、原始评分矩阵、维度限制 s.

输出:填充后的评分矩阵.

对于每个类别,分别进行下面的步骤:

Step 1. 随机初始化 $W \in R^+_{\|U(k)\| \times s}, H \in R^+_{s \times \|V(k)\|}$.

Step 2. 按照公式(1)调整 W.

Step 3. 按照公式(2)调整 H.

Step 4. 跳转 Step 2,直至收敛.

Step 5. 对于 $i \in U(k), j \in V(k)$, $R_{ij} = \begin{cases} R_{ij}, & \text{if } R_{ij} \neq 0 \\ W_i \times H_j, & \text{if } R_{ij} = 0 \end{cases}$.

算法 1 说明:第 1 阶段的 Step 4.2 是对 $p(r|k)$进行高斯平滑,没有 Step 4.2 的算法称为 Discrete BlockClust,进行了 Step 4.2 的高斯平滑之后的算法称为 Continuous BlockClust.

时间复杂度分析.第 1 阶段的时间复杂度是 $O(iter \times \| ratings \| \times \| K \|)$,第 2 阶段的时间复杂度是 $O(iter \times size(cluster) \times \| K \|)$.其中,$iter$ 是迭代次数,第 1 阶段在 20 以内,第 2 阶段为 200 左右.$size(cluster)$是类别内部非零元素的个数.由于可以限制类别内的评分数量规模,因此预测阶段的时间复杂度为常量级别,可以进行实时的更新.

在两阶段算法基础上,我们可以考虑利用 BlockClust 软聚类的优势.评分可能属于多个类别,并且评分属于这些类别的概率是不同的,因此可以有多个不同的预测结果.我们可以使用加权平均的方法来综合这些不同类别内的预测结果,从而达到改进预测结果的目的.具体操作见第 4.4 节.

3 增量式更新

在实际的推荐系统中,用户的兴趣经常发生变化,同时,用户随着使用的深入,期望系统做出越来越准确的推荐,这就要求推荐算法能够响应最新的训练数据,因此增量式的学习成为推荐系统的重要特性.遗憾的是,目前仅有少数协同过滤模型支持增量式的学习[22,23].

在本文提出的两阶段评分预测算法中,当我们得到了第 1 阶段的联合聚类结果后,原始评分矩阵被划分成为很多子矩阵,在这些子矩阵中,用户有着相似的兴趣,物品具有相似的属性,因此评分具有相似的模式.当有一些新的评分到达推荐系统时,我们只需要更新这些以前的未知项所处的类别,也即对这些子矩阵进行重新预测.推荐系统中的更新包括 3 种类型:新用户、新物品以及新评分[17].下面分别描述这 3 种情形对应的更新算法.

- 新用户 u_{new}:该用户对已存在的物品 v 进行了评分,记为 (u_{new},v,r).在物品 v 所属的类别内,补充评分(u_{new},v,r)对该类别进行重新预测,使用重新预测的结果进行推荐.
- 新物品 v_{new}:该物品被已存在的用户 u 给了一条评分,记为 (u,v_{new},r).在用户 u 所属的类别内,补充评分(u,v_{new},r)对该类别进行重新预测,使用重新预测的结果进行推荐.
- 新的评分$(u,v,r)_{new}$:已存在的用户 u 对于已存在的物品 v 给了一条新的评分,记为$(u,v,r)_{new}$.在用户 u 和物品 v 同时所属的类别内补充评分$(u,v,r)_{new}$,对该类别进行重新预测,使用重新预测的结果进行推荐.

除了根据新的评分数据在线调整类别内部预测之外,推荐系统应当定期对聚类结果进行调整,以便更加准确地反映最新的评分模式.

4 实验评价

4.1 电影评分数据集

Movielens(http://www.grouplens.org/node/73)是明尼苏达大学 GroupLens 小组搜集的电影评价数据集.该数据集包含了 6 040 位用户对 3 706 部电影的评分数据(1~5),评分总数为 1 000 000.其中评分最少的用户评分为 20 条,最多为 2 314 条;评分最少的电影的评分数是 1 条,最多的则有 3 428 条评分数据.总体数据稀疏度为 95.53%.该数据集的详细参数见表 2.

Table 2 Description of MovieLens movie rating dataset

表 2 MovieLens 电影评分数据集描述

User average rating number	Movie average rating number	Global mean rating	#1 point	#2 point	#3 point	#4 point	#5 point
166	270	3.58	56 161	107 534	261 156	348 883	226 266

我们在 MovieLens 数据集上进行了 4 组实验,分别验证本文提出的联合聚类算法效果、总体评分预测效果、考虑加权平均的改进效果以及增量式学习的效果.首先,我们抽取一个小规模数据集验证 BlockClust 聚类效果;然后在全体数据集上进行评分预测实验,并对比了相关方法的误差以及时间特性;在此基础上,由于 BlockClust 是软聚类方法,对评分所属类别内预测进行加权平均可以进一步改进预测效果;最后是面对 3 种更新模式,增量式更新算法的效果测试.

4.2 聚类效果评测

为了观察聚类效果,我们挑选了 MovieLens 中被评分最多的 10 部电影和评分最多的 10 个用户作为代表,其原始 id 以及相应的评分如图 3 所示(行对应用户,列对应电影).我们采用高斯平滑后的 BlockClust 算法,将用户和电影分别聚集到 5 个子类别中,每个类别具有相同或相似的评分模式.我们用不同的颜色代表不同的评分,具体的聚类结果如图 3 所示.在缩小评分矩阵规模的同时,提高了类别内部评分的统一性,从而提高了类别内部预测的准确度.

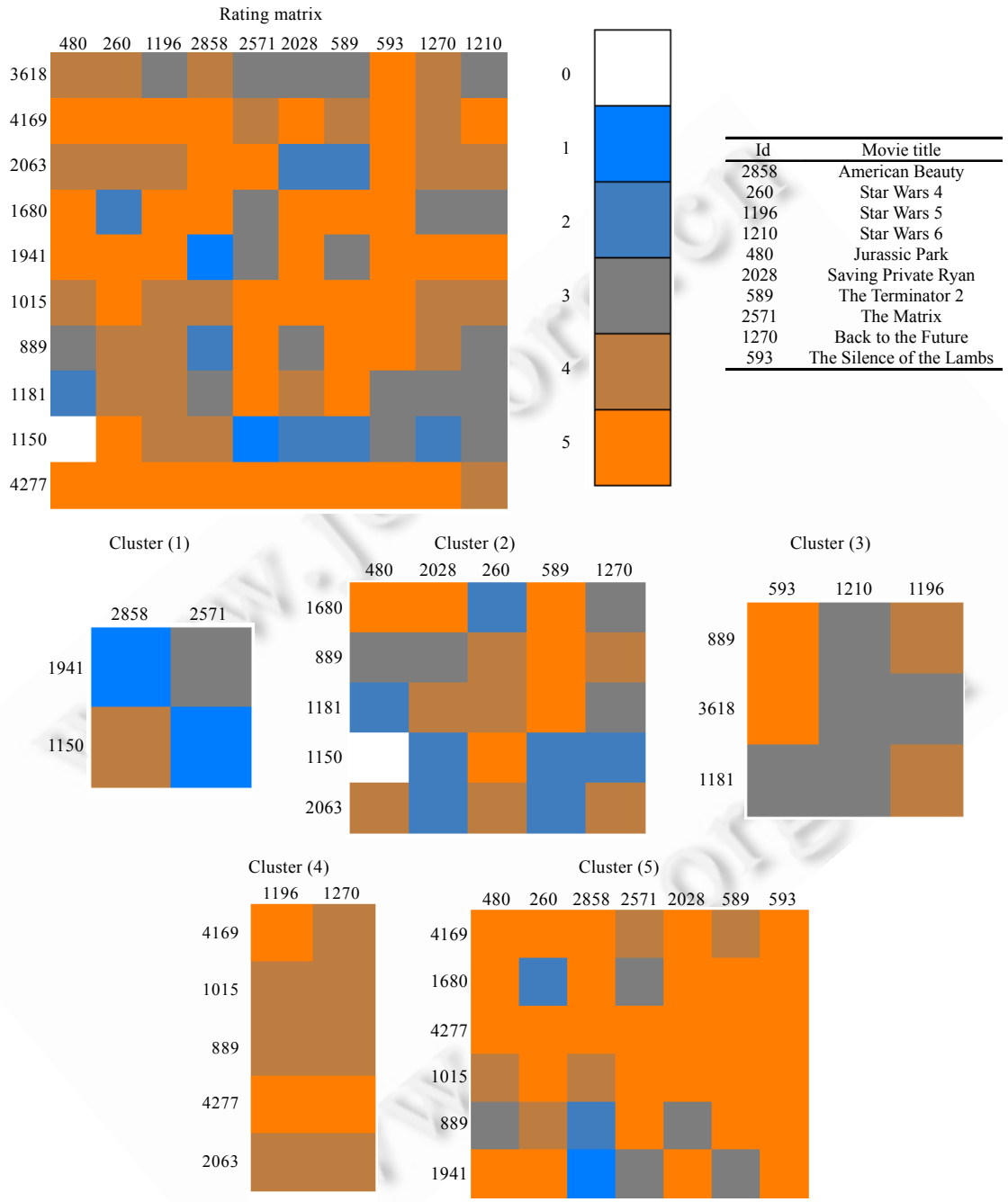


Fig.3 Hot movies clustering results

图 3 热门电影聚类效果

4.3 评分预测

我们采用均方根误差作为评价评分预测准确性的标准,也即对于未出现在训练集中但出现于测试集中的评分对计算 RMSE(root mean square error):

$$RMSE=\sqrt{\frac{1}{|TEST|}\sum_{(u,v,R_{u,v})\in TEST}(R_{u,v}-estimated)^2}.$$

参数设置.我们采用 BlockClust+NMF 对 MovieLens 进行评分预测.算法的超参数设置如下:$\alpha=\beta=\vartheta=0.00000001$.对于类别数的设置我们做了一组实验,对比结果见表 3(其中,实验硬件环境:1.66G 酷睿双核 CPU,3GB 内存;软件环境:Python 2.5,SciPy 0.70).当类别数为 50 时达到最佳效果.在以下的实验中,我们统一用 50 作为类别数.

Table 3 Relationship between cluster numbers and training time, RMSE

表 3 聚类类别数与训练时间以及 RMSE 的关系

Cluster number	20% rating training set		80% rating training set	
	RMSE	Time (s)	RMSE	Time (s)
1	1.124±0.03	421.1	1.131±0.01	630.3
5	1.120±0.003	432.5	1.111±0.02	487.4
10	1.117±0.008	**297.2**	1.081±0.04	423.6
30	1.106±0.04	398.8	1.054±0.05	**406.4**
50	**1.098±0.06**	414.6	**0.985±0.05**	432.4
100	1.152±0.07	584.1	1.032±0.06	593.2

横向对比.我们对比了以下 10 种方法:INMF,PLSA,Neighborhood,COCLUST,BregCoClust+NMF,BregCoClust+PLSA,BregCoClust+Neighborhood,我们提出的联合聚类方法的 3 种组合方法:BlockClust+INMF,BlockClust+PLSA,BlockClust+Neighborhood.其中,INMF 是经过我们在初始化阶段改进的文献[6]中的算法,PLSA 是文献[3]中的高斯分布平滑方法,Neighborhood 方法是我们实现的文献[24]中的近邻均值方法,COCLUST 是文献[17]中提出的直接通过聚类预测评分方法,BregCoClust 实现的是文献[13]中 C_5 联合聚类方法.对于本文提出的联合聚类方法,我们又实现了 Discrete BlockClust 和 Continuous BlockClust 两个版本,分别对应离散评分概率聚类和高斯平滑后的评分聚类.我们采取 5 次交叉验证(5-fold cross validation),即将 MovieLens 原始评分数据集划分为 5 份,每份数据集包含 20%的评分数据,每次使用其中 1 份数据作为训练集,其余作为测试集,重复 5 次;然后再将每份数据集分别作为测试集,而其余 80%评分作为训练集,也重复 5 次.这样,分别得到了 20%训练集和 80%训练集下各种算法的预测结果.实验结果对比见表 4.

Table 4 Comparison of RMSE for 10 algorithms on MovieLens dataset

表 4 10 种算法在 MovieLens 数据集上 RMSE 指标的横向比较

Algorithm		20% rating training set		80% rating training set	
		RMSE	Time (s)	RMSE	Time (s)
INMF		1.462±0.05	998.2	1.126±0.01	1721.1
PLSA		1.194±0.02	52712.8	1.093±0.02	112871.2
Neighborhood		1.392±0.02	**11.9**	1.371±0.03	**33.1**
COCLUST		2.069±0.05	36.0	1.873±0.05	115.6
BregCoClust+INMF		1.416±0.02	1241.5	1.201±0.02	1796.0
BregCoClust+PLSA		1.254±0.01	10254.2	1.276±0.008	28112.7
BregCoClust+Neighborhood		1.305±0.008	69.1	1.228±0.005	137.2
BlockClust+INMF	Discrete	1.098±0.04	303.1	**0.985**±0.05	409.6
	Continuous	**1.087**±0.06	414.6	1.011±0.03	610.7
BlockClust+PLSA	Discrete	1.114±0.02	3254.9	1.071±0.02	5610.1
	Continuous	1.121±0.01	3682.1	1.068±0.01	6294.6
BlockClust+Neighborhood	Discrete	1.144±0.03	92.5	1.107±0.02	204.3
	Continuous	1.143±0.02	94.7	1.112±0.03	215.4

实验结果分析.单个算法比较,Neighborhood 方法在时间性能上具有很大的优势,但准确性很差;在准确性指标上表现最好的是 PLSA,但时间消耗最高;通过第 1 阶段的联合聚类,我们的两阶段预测方法不仅在准确度指标上与 NMF,PLSA,Neighborhood 等方法相比有很大的提升,而且在时间指标上也要优于 NMF 和 PLSA 算法,预测阶段所耗费的时间与 Neighborhood 方法也是相当的.此外,COCLUST 和 Bregman 联合聚类方法在评分矩阵上的表现远远不够理想.从我们提出的 BlockClust 方法可以看出,进行高斯平滑带来的改进并不明显.

4.4 考虑隶属度的加权综合预测

由于 BlockClust 是一种软聚类方法,即每个评分可能属于多个类别,并且属于这些类别的概率有所差别.因此我们可以考虑综合这些类别各自的预测结果并结合相应的隶属度,从而改进最终的预测结果.假设评分 $\langle u,v,r\rangle$ 属于 S 个类别($C(u)\cap C(v)$),且隶属度分别为 $prob_k=\frac{p(k\mid u)p(k\mid v)}{\sum_{k'\in K}p(k'\mid u)p(k'\mid v)}\in[0,1],k\in[1,S]$, 预测结果分别为 $predict_k,k\in[1,S]$. 最终的预测结果为分类预测结果的加权平均,即 $predict=\frac{\sum_{k\in[1,S]}predict_k\times prob_k}{\sum_{k\in[1,S]}prob_k}$. 加权前(single)和加权后(bagging)的预测结果比较结果见表 5,加权后的预测效果在单个预测结果的基础上有进一步的提高(各实验环境下,RMSE 均有 2.74%~8.18%幅度的减小).

Table 5 Improvement of the weighted average of predictions from clusters

表 5 加权平均各类别预测结果带来的改进

Algorithm		20% rating training set		80% rating training set	
		RMSE	Time (s)	RMSE	Time (s)
Single	Discrete	1.098±0.04	**303.1**	0.985±0.05	**409.6**
	Continuous	1.087±0.06	414.6	1.011±0.03	610.7
Bagging	Discrete	1.023±0.04	321.5	**0.958**±0.04	427.1
	Continuous	**0.998**±0.03	432.1	0.971±0.03	616.3

4.5 增量式更新

我们挑选了 MovieLens 的一个子集进行静态模型的训练,静态模型包含 179 名用户对于 199 个物品的共 2 006 条评分.类似地,我们采用 5 次交叉验证的方法,训练方法包括了高斯平滑.在静态训练的基础上,我们分别测试了 3 种更新模式下我们的算法的表现:

- 新用户:在静态训练模型基础上,我们随机挑选了 10 名未参与训练的用户,他们分别对一个已有物品进行了评分.
- 新物品:类似地,我们向静态训练模型中增加了 10 个未参与训练的物品,它们分别被一些已存在的用户给出了 1 条评分.
- 新评分:这种更新情况最为常见.我们在测试集中选择了 100 条评分作为增量测试数据集,也即已存在用户对于已存在的物品新的评分.

表 6 是 COCLUST 方法与我们的联合聚类方法 BlockClust+INMF 增量学习效果的比较.其中,OLD RMSE 是增量更新前对相应项(用户或物品)未知评分的预测误差,NEW RMSE 是经过增量更新后的预测误差.我们的算法在处理新物品和新评分方面具有很大的优势(对于新评分,COCLUST 算法不作更新).对于新用户的情形,当一个新用户对某个物品进行了评分,也即该用户仅有 1 个有效评分.由于 COCLUST 使用已知物品的平均评分进行预测,因此不受用户评分数量的影响,效果反而比 BlockClust 利用这唯一的评分来进行预测更好.在训练时间方面,两个算法都是在常量时间内完成更新.

Table 6 Comparison of the incremental learning performance

表 6 增量学习效果对比

Incremental scenario	COCLUST			BlockClust+NMF		
	Time (s)	OLD RMSE	NEW RMSE	Time (s)	OLD RMSE	NEW RMSE
New user	**0.023**	/	**1.143**±0.08	0.054	2.356±0.10	1.315±0.04
New item	**0.271**	/	1.462±0.10	0.312	2.359±0.09	**1.174**±0.03
New rating	/	/	/	0.025	0.983±0.05	**0.702**±0.04

5 结论与未来的工作

本文提出了一种两阶段评分预测方法,其流程是先对评分矩阵进行联合聚类,再进行类别内部的评分预测.

联合聚类阶段,为了更加精确地反映评分特征,我们不能简单地将用户或者物品划归某个类别,而是采用了对用户和物品进行联合聚类的方法,即以评分值为标准,寻找具有相同模式的评分块,从而把原始评分矩阵划分成为相互可能有交叉的评分子块,也就是算法所获取的类别.在评分预测阶段,我们对这些子类进行加权的非负矩阵分解 NMF 操作,可以获得对其中未知项的预测.由于算法控制了类别内部评分数量的规模,因此 NMF 可以很快地完成.

当有新的评分加入评分矩阵时,更新算法查找对应的评分属于哪个或哪些类别,通过更新这些类别内部的预测,我们可以快速地完成模型的更新.实验结果表明,无论是评分的准确性还是算法更新的效率,本文提出的两阶段评分预测方法都要优于其他相关方法或者它们的组合,可以应用于大规模实时推荐系统.

进一步考虑到在联合聚类步骤中算法能够使用的信息是非常稀疏的评分矩阵,不利于准确地聚类,一个很自然的改进方法是引入迭代:在每个循环内首先进行联合聚类,在聚类的基础上进行评分预测,然后再使用评分预测的结果重新进行聚类,如此循环直至收敛.Pan 等人[25]也提出了一种类似的循环迭代的两阶段联合聚类框架,所不同的是,其中的矩阵分解仅仅为了降低数据维度,而我们使用矩阵分解的方法是为了预测类别内的未知评分.由于预测得到的评分可靠度不如原始评分高,在迭代的过程中,联合聚类算法必须考虑到评分的权重,这样才能保证聚类结果的可靠性.这就要求更加灵活的加权联合聚类算法,这是我们进一步研究的方向.

References:

[1] Xu HL, Wu X, Li XD, Yan BP. Comparison study of Internet recommendation system. Journal of Software, 2009,20(2):350−362 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3388.htm [doi: 10.3724/SP.J.1001.2009.03388]

[2] Marlin B. Collaborative Filtering: A machine learning perspective [MS. Thesis]. Toronto: University of Toronto, 2004.

[3] Hofmann T. Latent semantic models for collaborative filtering. ACM Trans. on Information System, 2004,22(1):89−115. [doi: 10.1145/963770.963774]

[4] Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. Journal of Machine Learning Research, 2003,3(3):993−1022. [doi: 10.1162/ jmlr.2003.3.4-5.993]

[5] Netflix update: Try this at home. 2006. http://sifter.org/~simon/journal/20061211.html

[6] Zhang S, Wang WH, Ford J, Makedon F. Learning from incomplete ratings using non-negative matrix factorization. In: Ghosh J, ed. Proc. of the 6th SIAM Conf. on Data Mining. Bethesda: SIAM, 2006. 549−553.

[7] Cheng YZ, Church GM. Biclustering of expression data. In: Bourne PE, ed. Proc. of the 8th Int'l Conf. on Intelligent Systems for Molecular Biology. La Jolla: AAAI Press, 2000. 93−103. [doi: 10.1016/j.ipm.2008.12.004]

[8] Cheng G, Wang F, Zhang CS. Collaborative filtering using orthogonal nonnegative matrix tri-factorization. Information Processing & Management, 2009,45(3):368−379.

[9] Shan HH, Banerjee A. Bayesian co-clustering. In: Altman R, ed. Proc. of the ICDM 2008. Washington: IEEE Computer Society Press, 2008. 530−539.

[10] Dhillon SI. Co-Clustering documents and words using bipartite spectral graph partitioning. In: Lee D, ed. Proc. of the 7th ACM SIGKDD. New York: ACM Press, 2001. 269−274.

[11] Wang J, de Vries AP, Reinders MJT. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Efthimiadis EN, ed. Proc. of the 29th Annual Int'l ACM SIGIR. New York: ACM Press, 2006. 501−508.

[12] Dhillon IS, Mallela S, Modha DS. Information-Theoretic co-clustering. In: Getoor L, ed. Proc. of the 9th ACM SIGKDD. New York: ACM Press, 2003. 89−98.

[13] Banerjee A, Dhillon I, Ghosh J, Merugu S, Modha DS. A generalized maximum entropy approach to Bregman co-clustering and matrix approximation. Journal of Machine Learning Research, 2007,8(8):1919−1986.

[14] Agarwal D, Merugu S. Predictive discrete latent factor models for large scale dyadic data. In: Berkhin P, ed. Proc. of the SIGKDD. New York: ACM Press, 2007. 26−35.

[15] Li XG, Yu G, Wang DL, Bao YB. Latent concept extraction and text clustering based on information theory. Journal of Software, 2008,19(9):2276−2284 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/2276.htm [doi: 10.3724/SP.J.1001.2008.02276]

[16] Shafiei MM, Milios EE. Latent Dirichlet co-clustering. In: Liu JM, ed. Proc. of the 6th Int'l Conf. on Data Mining. Washington: IEEE Computer Society Press, 2006. 542−551.

[17] George T, Merugu S. A scalable collaborative filtering framework based on co-clustering. In: Raghavan V, ed. Proc. of the 5th IEEE Int'l Conf. on Data Mining. Washington: IEEE Computer Society Press, 2005. 625−628.

[18] Long B, Zhang ZF, Yu PS. Co-Clustering by block value decomposition. In: Grossman R, ed. Proc. of the SIGKDD 2005. New York: ACM Press, 2005. 635−640.

[19] Gaussier E, Goutte C. Relation between PLSA and NMF and implications. In: Marchionini G, ed. Proc. of the 28th Annual Int'l ACM SIGIR. New York: ACM Press, 2005. 601−602.

[20] Donoho D, Stodden V. When does non-negative matrix factorization give a correct decomposition into parts? In: Thrun S, Saul L, Schölkopf B, eds. Advances in Neural Information Processing Systems 16. Cambridge: MIT Press, 2004. 1141−1148.

[21] Langville AN, Meyer CD, Albright R. Initializations for the nonnegative matrix factorization. In: Ungar L, ed. Proc. of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2006. http://meyer.math.ncsu.edu/Meyer/PS-Files/NMFInit.pdf

[22] Cao B, Shen D, Sun JT, Wang XH, Yang Q, Chen Z. Detect and track latent factors with online nonnegative matrix factorization. In: Proc. of the IJCAI. 2007. 2689−2694. http://dli.iiit.ac.in/ijcai/IJCAI-2007/PDF/IJCAI07-432.pdf

[23] Wu H, Zhang D, Wang YJ, Cheng X. Incremental probabilistic latent semantic analysis for automatic question recommendation. In: Pu P, ed. Proc. of the Recommender System 2008. New York: ACM Press, 2008. 99−106.

[24] Koren Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In: Li Y, ed. Proc. of the 14th ACM SIGKDD. New York: ACM Press, 2008. 426−434.

[25] Pan F, Zhang X, Wang W. A general framework for fast co-clustering on large datasets using matrix decomposition. In: Alonso G, ed. Proc. of the 24th Int'l Conf. on Data Engineering. Washington: IEEE Computer Society Press, 2008. 1337−1339.

附中文参考文献:

[1] 许海玲,吴潇,李晓东,阎保平.互联网推荐系统比较研究.软件学报,2009,20(2):350−362. http://www.jos.org.cn/1000-9825/3388.htm [doi: 10.3724/SP.J.1001.2009.03388]

[15] 李晓光,于戈,王大玲,鲍玉斌.基于信息论的潜在概念获取与文本聚类.软件学报,2008,19(9):2276−2284. http://www.jos.org.cn/1000-9825/19/2276.htm [doi: 10.3724/SP.J.1001.2008.02276]

吴湖(1982－),男,安徽望江人,博士,CCF学生会员,主要研究领域为协同过滤算法.

王永吉(1962－),男,博士,研究员,博士生导师,CCF高级会员,主要研究领域为计算机实时系统,网络与通信技术,数据挖掘,智能软件工程.

王哲(1985－),男,硕士生,主要研究领域为推荐系统.

王秀利(1977－),男,博士,讲师,主要研究领域为可信计算,计算机网络,优化理论及应用.

杜栓柱(1971－),男,博士,高级工程师,主要研究领域为业务过程建模理论与技术,知识工程.