基于用户谱聚类的协同过滤推荐算法

李振博,徐桂琼,查 九
(上海大学 管理学院,上海 200444)

摘 要:针对电子商务系统中传统协同过滤推荐算法面临的稀疏性、准确性、实时性等问题,提出了一种基于用户谱聚类的协同过滤推荐算法。首先利用非负矩阵分解的方法对原始稀疏评分矩阵进行平滑处理,然后利用改进相似度的谱聚类方法将用户聚类,最后在用户所属类中寻找最近邻并产生推荐。用户谱聚类过程可离线完成,加快了在线推荐速度。在数据集 MovieLens 上的实验结果表明,该算法在平均绝对偏差、召回率、准确率等方面都有了较大改善,提高了推荐质量。

关键词:协同过滤;非负矩阵分解;相似度;谱聚类
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2014)09-0059-04
doi:10.3969/j.issn.1673-629X.2014.09.013

A Collaborative Filtering Recommendation Algorithm Based on User Spectral Clustering

LI Zhen-bo, XU Gui-qiong, ZHA Jiu
(School of Management, Shanghai University, Shanghai 200444, China)

Abatract: Considering the sparsity, accuracy and the real-time problem of traditional collaborative filtering recommendation algorithms in electronic commerce system, a new collaborative filtering algorithm based on user spectral clustering is proposed. Firstly, it employs non-negative matrix factorization algorithm to fill the missing ratings. Then, it uses spectral clustering method of improved similarity to cluster users. Finally, it finds the nearest neighbors of the user according to the user's cluster and generates recommendations. Spectral clustering can be performed by off-line, which will accelerate the speed of online recommendation. The experimental results on MovieLens show that the new algorithm improves recommendation quality in MAE, recall and precision.

Key words: collaborative filtering; Non-negative Matrix Factorization (NMF); similarity; spectral clustering

0 引 言

协同过滤算法是推荐系统中应用最广泛,效果最好的方法之一。它的基本思想是利用用户的历史喜好信息来计算用户之间的距离,然后利用目标用户的"最近邻居"对商品的评价来预测目标用户对特定商品的喜好程度,根据此喜好程度来对目标用户进行推荐[1]。其优点是对所推荐的资源类型没有特殊要求,可以处理非结构化的复杂对象[2]。但是随着电子商务网站用户和资源数量不断的增加,传统的协同过滤算法面临数据稀疏、实时性和可扩展性等问题,推荐质量难以保证。

为了缓解上述问题,许多学者进行了深入研究并取得了一定的研究成果,如在传统相似性度量方法的基础上提出了改进的相似性计算方法[3-5],提高了推荐准确度;将 SVD[6-7]、NMF[8-9]等矩阵分解技术应用于协同过滤算法,有效缓解了数据稀疏问题;将聚类技术引入协同过滤中有效提高了系统实时性。文献[10-12]使用 K-means 方法对用户或是项目聚类,减少了寻找最近邻的开销。但是,传统聚类方法存在对数据集的空间分布敏感等缺点,聚类效果并不理想。

谱聚类作为一种 K-means 的改进算法,具有收敛于全局最优解,实现简单等优点,成为近年来的研究热点,但在推荐系统中的应用较少。文中将谱聚类算法应用到推荐系统,提出了一种基于用户谱聚类的协同过滤推荐算法。

收稿日期:2013-11-11 **修回日期**:2014-02-15 **网络出版时间**:2014-07-17
基金项目:国家自然科学基金资助项目(11201290)
作者简介:李振博(1985-),女,硕士研究生,研究方向为数据挖掘、个性化推荐;徐桂琼,博士,教授,研究方向为复杂系统建模与动力学研究、数据挖掘、协同过滤。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20140717.1229.025.html

1 协同过滤算法概述

在一个基于协同过滤算法的推荐系统中,输入数据为用户评分矩阵,记作 $\boldsymbol{R}(m,n)$,如表 1 所示,其中 R_{ij} 代表用户 i 对项目 j 的评分。

表 1 用户评分矩阵

	Item_1	…	Item_j	…	Item_n
User_1	R_{11}	…	R_{1j}	…	R_{1n}
…	…	…	…	…	…
User_i	R_{i1}	…	R_{ij}	…	R_{in}
…	…	…	…	…	…
User_m	R_{m1}	…	R_{mj}	…	R_{mn}

传统的协同过滤算法首先是根据用户评分矩阵计算用户之间的相似度。计算两个用户之间的相似度主要有 3 种方法:余弦相似性、修正余弦相似性、Pearson 相关相似性。用户 u 和 v 之间的相似度记为 $\text{sim}(u,v)$。设 R_{ui}、R_{vi} 分别表示用户 u、v 对项目 i 的评分。用户 u、v 在项目集 I 上共同评分的项目集为 $I_{uv}=\{i\in I \mid R_{ui}\neq 0 \cap R_{vi}\neq 0\}$($I$ 为全部项目集)。则最常用的 Pearson 相关相似性可由公式(1)计算得到:

$$\text{sim}(u,v)=\frac{\sum_{i\in I_{uv}}(R_{ui}-\overline{R_u})(R_{vi}-\overline{R_v})}{\sqrt{\sum_{i\in I_{uv}}(R_{ui}-\overline{R_u})^2}\sqrt{\sum_{i\in I_{uv}}(R_{vi}-\overline{R_v})^2}} \tag{1}$$

其中,$\overline{R_u}$ 表示用户 u 的平均评分;$\overline{R_v}$ 表示用户 v 的平均评分。

然后根据计算得到的相似度,确定目标用户的最近邻居。对目标用户而言,在整个评分矩阵空间中搜索出 m 个相似度最高的用户即可以组成其最近邻居集合 $V=\{v_1,v_2,\cdots,v_m\}$。

最后根据当前用户最近邻居对项目的评分信息预测当前用户对其未评分项目的评分,以此产生 Top-N 推荐。用户 u 对未评分项目 i 的预测评分 P_{ui} 可以通过用户 u 的最近邻居集合 V 对 i 的评分得到,计算方法如公式(2):

$$P_{ui}=\overline{R_u}+\frac{\sum_{v\in V(u)}\text{sim}(u,v)\cdot(R_{vi}-\overline{R_v})}{\sum_{v\in V(u)}|\text{sim}(u,v)|} \tag{2}$$

2 基于用户谱聚类的协同过滤推荐算法

2.1 谱聚类算法

谱聚类算法是一种基于图论的聚类算法,与传统的聚类算法相比,谱聚类算法具有明显的优势,不仅实现简单,而且能够识别任意形状的样本空间,且能收敛于全局最优解,非常适合于实际问题。它最初被用于语音识别、图像分割、VLSI 设计等领域,近几年开始用于机器学习中,并迅速成为国际上机器学习领域的研究热点[13]。

谱聚类的思想来源于谱图划分,其本质是将聚类问题转化为图的最优划分问题[14]。将每个数据样本看作图中的顶点 V,根据样本间的相似度将顶点间的边 E 赋权重值 w,这样就得到一个基于样本相似度的无向加权图 $G(V,E)$,那么在图 G 中,就可将聚类问题转化为在图 G 上的图划分问题。谱聚类算法的主要工具是图的拉普拉斯矩阵。在无向加权图 G 上,定义其权值矩阵为 $\boldsymbol{W}$,则图 G 的未规范的拉普拉斯矩阵 $\boldsymbol{L}$ 如公式(3)所示:

$$\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W} \tag{3}$$

其中,$\boldsymbol{D}$ 为对角矩阵,D_{ii} 为 $\boldsymbol{W}$ 矩阵第 i 行元素之和。

图 G 的规范的拉普拉斯矩阵 $\boldsymbol{L}_{\text{sym}}$ 如公式(4)所示:

$$\boldsymbol{L}_{\text{sym}}=\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}=\boldsymbol{I}-\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2} \tag{4}$$

根据不同的图的划分准则函数及谱映射方法,谱聚类算法有着不同的具体实现方法。总的来说可归纳为以下三个步骤:

(1)对数据集进行预处理,构建相似度矩阵 $\boldsymbol{W}$;

(2)构造拉普拉斯矩阵,选取合适的特征向量,构建特征向量空间;

(3)利用 K-means 聚类算法对特征向量空间中的特征向量进行聚类。

2.2 改进的相似度度量方法

经典的谱聚类算法常用高斯核函数作为相似性度量的方法,虽然该函数使原始的谱聚类算法取得了一些成果,但尺度参数 σ 的选取问题使该函数具有明显的局限性。为此,文中使用改进的 Pearson 相关系数作为相似性度量方法。

协同过滤算法中常用的 Pearson 相关相似性度量方法是基于用户共同评分的。通常情况下,用户的共同评分项目非常少,但是由于他们的评分非常接近,因此计算的相似性很高。实际上这种情况可能会高估用户的相似性,如果两个用户同时感兴趣的项目个数很少,那么他们的兴趣爱好可能很不相似。所以,在选择共同评分数据的前提下,考虑共同评分的项目个数,才能更准确地表示用户兴趣爱好的相似程度。为此,引入 Salton 系数,对过度估计的 Pearson 相似度进行修正。Salton 系数计算方法如公式(5)所示:

$$\text{sim}(u,v)=\frac{|N(u)\cap N(v)|}{\sqrt{|N(u)|}\sqrt{|N(v)|}} \tag{5}$$

其中,$N(u)$ 表示用户 u 的评分项集合;$N(v)$ 表示

用户 v 的评分项集合。

用户 u 和用户 v 的 Pearson 相关系数记为 P_{uv}, Salton 系数记为 S_{uv},将 Pearson 相关系数与 Salton 系数相结合,得到用户 u 和用户 v 的改进相似度 PS_{uv},其表达式如公式(6)所示:

$$\mathrm{PS}_{uv} = P_{uv} \times S_{uv} \tag{6}$$

2.3　基于用户谱聚类的协同过滤推荐算法步骤

基于用户谱聚类的协同过滤推荐算法(NPSSC)可分为离线谱聚类和在线 Top-N 推荐两个阶段。谱聚类算法有着不同的具体实现方法,文中选用谱聚类中常用的 NJW 算法。

NPSSC 算法具体步骤描述如下:

离线谱聚类阶段:

(1) 利用NMF技术对原始评分矩阵 $\boldsymbol{R}$ 进行平滑处理,得到填充后的矩阵 $\boldsymbol{R}_{\mathrm{fill}}$;

(2) 根据改进相似度公式(6),计算用户相似度矩阵 $\boldsymbol{PS}$,作为谱聚类的输入矩阵 $\boldsymbol{W}$;

(3) 根据公式(4) 构造规范化的拉普拉斯矩阵 $\boldsymbol{L}_{\mathrm{sym}}$;

(4) 对 $\boldsymbol{L}_{\mathrm{sym}}$ 进行特征分解,选取前 k 个特征值对应的特征向量 $\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_k$ 组成矩阵 $\boldsymbol{V} \in R^{n \times k}$ 并对 $\boldsymbol{V}$ 进行规范化处理;

(5) 利用 K - means 聚类方法将特征向量空间矩阵 $\boldsymbol{V}$ 聚为 k 类,即将用户聚为 k 类,表示为 $C_1, C_2, \cdots, C_k$。

在线 Top - N 推荐阶段:

(1) 在用户所属类中寻找用户最近邻,生成用户近邻集合;

(2) 利用公式(2) 计算用户未评分项目的预测值,生成用户预测评分矩阵 $\boldsymbol{R}_{\mathrm{pre}}$;

(3) 对 $\boldsymbol{R}_{\mathrm{pre}}$ 进行排序,将预测评分最高的 Top - N 个项目推荐给用户。

3　实验及结果分析

3.1　实验数据集

采用 MovieLens 数据集作为实验数据集。该数据集由美国的 GroupLens 研究小组提供,包含了 943 个用户对 1 682 部电影在连续 7 个月内的 10 万条评分数据。用户评分值用 1-5 之间的整数表示,数值越大,说明用户对项目的喜爱程度越高;反之,说明用户对项目兴趣度不高。数据集的稀疏度(未知评分在整个数据集所占的比例)为:1-100 000/943×1 682=93.7%。

3.2　评价指标

(1)平均绝对偏差(Mean Absolute Error,MAE),即预测的用户评分与实际的用户评分的偏差。MAE 越小说明推荐准确度越高。在测试集 T 上,MAE 定义为公式(7):

$$\mathrm{MAE} = \frac{\sum_{u,i \in T} | r_{ui} - p_{ui} |}{|T|} \tag{7}$$

(2)Top-N 推荐的准确性指标:召回率(recall)、准确率(precision)。设 $R(u)$ 是根据用户在训练集上的行为给用户做出的推荐列表,而 $T(u)$ 是用户在测试集上的行为列表。推荐结果的召回率定义为公式(8):

$$\mathrm{Recall} = \frac{\sum_{u \in U} | R(u) \cap T(u) |}{\sum_{u \in U} | T(u) |} \tag{8}$$

推荐结果的准确率定义为公式(9):

$$\mathrm{Precision} = \frac{\sum_{u \in U} | R(u) \cap T(u) |}{\sum_{u \in U} | R(u) |} \tag{9}$$

3.3　实验结果分析

随机抽取数据集的 80% 作为训练集,其余 20% 作为测试集。实验可分为以下两个部分:

(1)预测评分的准确性实验。

由于 NPSSC 中聚类的个数对算法的性能有重要影响,聚类数目过多则每个簇中用户数就过少,用户近邻数也过少,影响预测评分准确性;聚类数目过少则每个簇中用户数就过多,算法的实时性得不到改善。所以,首先对聚类个数进行最优化实验。将聚类数目范围定为5 到15,分别计算对应的 MAE 值,确定 MAE 值最小的为 NPSSC 算法聚类数目。实验结果如图 1 所示。

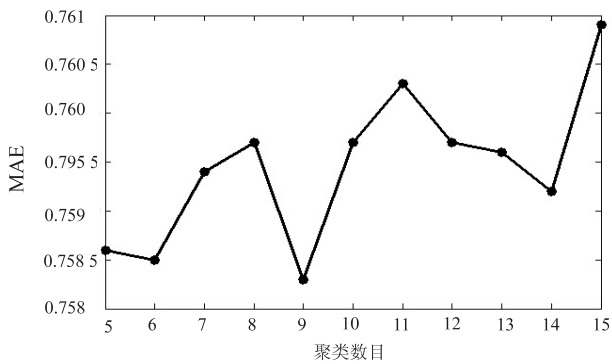

图 1　不同聚类数目 NPSSC 算法 MAE 比较

由图 1 可知:聚类数目为 9 时,NPSSC 算法 MAE 值最小。由表 2 可知:各簇中用户数分布比较均匀,谱聚类的结果比较合理。

表 2　不同簇中用户数

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
E_1	61	86	88	97	99	103	131	138	140

确定最佳聚类数目后,比较各算法 MAE 值大小。P 代表传统的基于用户 Pearson 相似度的协同过滤算法,PS 代表将用户 Pearson 相似度和 Salton 相结合的

协同过滤算法,NMF 代表基于非负矩阵分解的协同过滤算法,NPSSC 代表文中提出的基于用户谱聚类的协同过滤推荐算法。结果如图 2 所示,P 算法 MAE 值最大,PS 算法 MAE 值大于 NMF 算法,NPSSC 算法 MAE 值最小。

实验结果表明,NPSSC 算法与 P、PS、NMF 算法相比,在 MAE 指标上有显著的降低,大大提高了评分预测准确度。

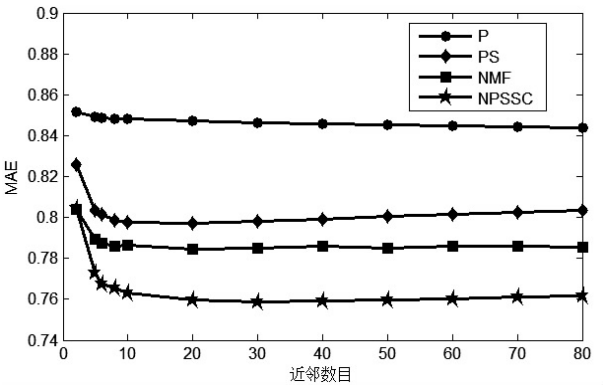

图 2 不同协同过滤算法 MAE 比较

(2) Top-N 推荐的准确性实验。

选择推荐列表长度为 20、30、40 时进行各算法召回率和准确率的实验。图中四种不同柱状图从左到右依次代表 P、NMF、PS、NPSSC 四种算法。实验结果如图 3、4 所示。

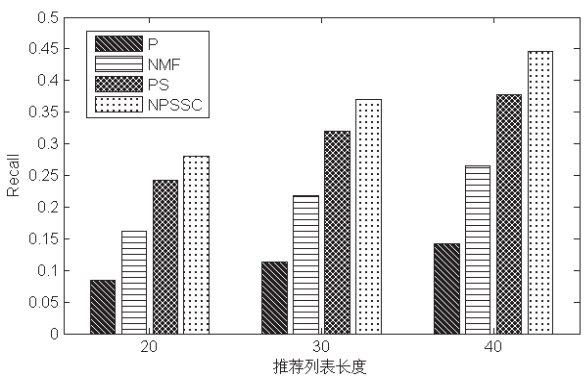

图 3 不同协同过滤算法召回率比较

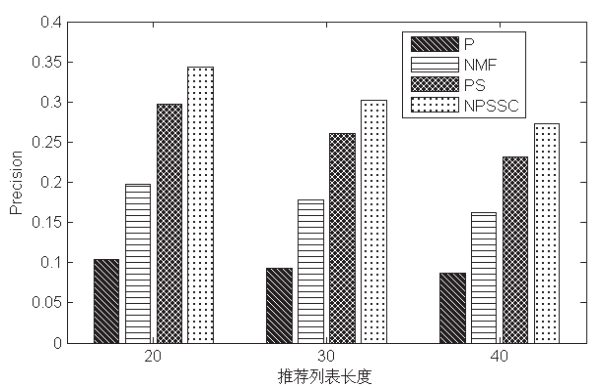

图 4 不同协同过滤算法准确率比较

由图 3 可知,随着推荐列表长度的增加,各算法的召回率都有所提高。其中 P 算法召回率最低,NMF 算法召回率低于 PS 算法召回率,NPSSC 算法召回率最高。由图 4 可知,随着推荐列表长度的增加,各算法的准确率都有所降低。其中 P 算法准确率最低,NMF 算法准确率低于 PS 算法准确率,NPSSC 算法准确率最高。

实验结果表明,NPSSC 算法比 P、NMF、PS 算法在召回率和准确率指标上有了较大的提高,有效改善了推荐算法的性能。

4 结束语

文中提出的基于用户谱聚类的协同过滤推荐算法(NPSSC),首先使用 NMF 技术对原始评分矩阵进行平滑处理,解决了原始评分矩阵的稀疏问题;然后用改进的相似性度量方法计算用户的相似度,再对用户进行谱聚类,避免了谱聚类算法中相似度的尺度参数的问题,提高了推荐的准确性;最后在用户所属类别中寻找用户最近邻,并产生推荐结果,降低了在线计算的向量空间维度,从而大大减少了在线算法的执行时间,提高了协同过滤算法的实时性。

在 MovieLens 数据集上的实验结果表明,该算法在 MAE、召回率和准确率指标上都有了较大的改善。文中聚类个数是通过实验事先确定的,如何自动确定最佳聚类数目,是值得进一步研究的问题。

参考文献:

[1] 奉国和,梁晓婷. 协同过滤推荐研究综述[J]. 图书情报工作,2011,55(16):126-130.

[2] 许海玲,吴 潇,李晓东,等. 互联网推荐系统比较研究[J]. 软件学报,2009,20(2):350-362.

[3] Jeong B,Lee J,Cho H. Improving memory-based collaborative filtering via similarity updating and prediction modulation[J]. Information Sciences,2010,180(5):602-612.

[4] Zhao Changwei,Peng Qinke,Liu Che. An improved structural equivalence weighted similarity for recommender systems[J]. Procedia Engineering,2011,15:1869-1873.

[5] 李克潮,蓝冬梅. 一种属性和评分的协同过滤混合推荐算法[J]. 计算机技术与发展,2013,23(7):116-119.

[6] Vozalis M G,Margaritis K G. Using SVD and demographic data for the enhancement of generalized collaborative filtering [J]. Information Sciences,2007,177(15):3017-3037.

[7] 杨 阳,向 阳,熊 磊. 基于矩阵分解与用户近邻模型的协同过滤推荐算法[J]. 计算机应用,2012,32(2):395-398.

[8] Chen Gang,Wang Fei,Zhang C S. Collaborative filtering using orthogonal nonnegative matrix tri-factorization[J]. Information Processing and Management,2009,45(3):368-379.

(下转第 67 页)

-本地调度-CPU 调度,形成了不同层次的调度。作业调度的流程设计描述如下:

(1)接收浏览器端用户提交的作业请求描述和调度说明,组装成作业对象,加入到就绪队列之中。

(2)通过监控信息系统得到系统运行状态,提供资源的使用状况及所运行作业的状态查询功能。

(3)分析作业请求描述及调度说明,匹配可用资源,得到候选资源集合。

(4)根据调度策略调度算法实现作业到特定集群资源的匹配,将作业转发到消息服务器上,同时作业从就绪队列中出队,加入到运行队列中。

(5)相应的本地守护进程从消息服务器取出作业请求描述消息,提交给本地作业调度器,并最终将作业分配到具体的处理机上运行。

4.2.5　作业控制及状态管理模块设计

作业控制、作业状态管理也是作业管理的重要组成部分。作业控制消息发送到消息服务器,相应的本地集群代理程序获取到该控制消息,并提交本地作业管理软件。

本地集群代理程序定期向本地作业管理软件发送作业状态查询请求,如果发现作业状态发生改变,将状态消息发送到消息服务器,全局作业管理器获取到该消息,解析该消息,将相应作业从运行队列中取出,加入到完成队列。

5　结束语

文中结合中国教育科研网格材料高性能计算服务门户系统项目建设背景,就如何整合不同时期、不同地域建设的高性能计算集群,提高各计算集群资源利用率,使用户可以通过提供的统一平台访问和使用这些高性能计算资源展开讨论。设计并实现了一种采用消息模型的多集群作业管理方案。该方案可以根据用户作业的资源需求及各个集群自身负载情况进行统一管理和调度。笔者基于这个原理设计并实现了该原型系统。采用该方案设计实现的多集群任务管理系统性能稳定,能实现多集群资源监控、资源管理、作业调度、作业控制、数据管理等功能。跨集群作业管理实现了不同集群利用率均衡的同时,有效解决了在资源异构及网络环境不可靠条件下的系统稳定性问题,显著提高了多集群系统作业吞吐能力。

参考文献:

[1] Buyya R. 高性能集群计算:结构与系统(第一卷)[M]. 英文版. 北京:人民邮电出版社,2002.

[2] 张小林,钟亦平. 基于集群系统的资源管理系统的性能分析与比较[J]. 计算机应用研究,2003,20(9):56-59.

[3] 雷　州,徐志伟,祝明发. 机群管理系统的比较与评价[J]. 计算机科学,1999,26(8):23-26.

[4] LSF Team. LSF administrator's guide[EB/OL]. 2007. http://www. platform. com/ products.

[5] Bayucan A. Portable batch system open PBS release 2. 3 administrator guide[EB/OL]. 2000. http://www. pbspro. com.

[6] 李全枝,梁正友. 集群资源管理系统 PBS 及其应用[J]. 微机发展(现更名:计算机技术与发展),2005,15(4):4-7.

[7] 李　源,郑全录,曾　韵. PBS 作业管理系统分析[J]. 现代计算机:上下旬,2004(3):17-19.

[8] He Ligang, Jarvis S A, Spooner D P, et al. Dynamic scheduling of parallel jobs with QoS demands in multiclusters and grids[C]//Proc of fifth IEEE/ACM international workshop on grid computing. [s. l.]:[s. n.],2004:402-409.

[9] 王　军. 基于 JMS 的消息中间件设计与实现[J]. 计算机应用,2003,23(8):64-67.

[10] 朱方娥,曹宝香. 基于 JMS 的消息队列中间件的研究与实现[J]. 计算机技术与发展,2008,18(5):172-175.

[11] Schwarzkopf M, Konwinski A. Omega: flexible, scalable schedulers for large compute clusters[J]. ACM Transactions on Computer Systems,2008,26(4):24-26.

[12] Cao Junwei, Spooner D P, Turner J D, et al. Agent-based resource management for grid computing[C]//Proc of 2nd IEEE/ACM symposium on cluster computing and the grid. [s. l.]:IEEE,2002:350-350.

[13] Li Hui, Groep D, Wolters L. Workload characteristics of a multi-cluster supercomputer[C]//Proceedings of the 10th international conference on job scheduling strategies for parallel processing. [s. l.]:[s. n.],2004:176-193.

[14] 武　斌. 网格市场环境下资源调度机制研究[D]. 合肥:中国科学技术大学,2010.

(上接第 62 页)

[9] 周子亮. 结合非负矩阵分解的推荐算法及框架研究[D]. 北京:北京交通大学,2013.

[10] 郁　雪,李敏强. 一种结合有效降维和 K-means 聚类的协同过滤推荐模型[J]. 计算机应用研究,2009,26(10):3718-3720.

[11] 徐　红,彭　黎,郭艾寅,等. 基于用户多兴趣的协同过滤策略改进研究[J]. 计算机技术与发展,2011,21(4):73-76.

[12] Tsi C F, Hung C. Cluster ensembles in collaborative filtering recommendation[J]. Applied Soft Computing,2012,12(4):1417-1425.

[13] 蔡晓妍,戴冠中,杨黎斌. 谱聚类算法综述[J]. 计算机科学,2008,35(7):14-18.

[14] von Luxburg U. A tutorial on spectral clustering[J]. Statistics and Computing,2007,17(4):395-416.

基于用户谱聚类的协同过滤推荐算法

作者： 李振博， 徐桂琼， 查九， LI Zhen-bo， XU Gui-qiong， ZHA Jiu
作者单位： 上海大学 管理学院,上海,200444
刊名： 计算机技术与发展 ISTIC
英文刊名： Computer Technology and Development
年，卷(期)： 2014(9)