第 36 卷 第 6 期 计 算 机 工 程 2010 年 3 月
Vol.36 No.6 Computer Engineering March 2010
·软件技术与数据库· 文章编号：1000—3428(2010)06—0052—03 文献标识码：A 中图分类号：TP391

# 协同过滤算法中的相似度优化方法

徐 翔，王煦法
(中国科学技术大学计算机科学与技术系，合肥 230027)

**摘 要**：在协同过滤推荐系统中，通过对稀疏评分矩阵进行填充，可以提高对用户相似度的度量效果和系统的推荐精度。不同填充方法对相似度计算结果的影响存在较大差异。为解决该问题，针对 3 类填充方法构建的评分数据集，以最近邻算法进行推荐，分析传统相似度和基于云模型的相似度经 2 种方法优化后的度量效果，分别为各填充方法选取最有效的相似度优化方案。

**关键词**：协同过滤；最近邻；相似度；云模型

# Optimization Method of Similarity Degree in Collaborative Filter Algorithm

XU Xiang, WANG Xu-fa
(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027)

**【Abstract】** In collaborative filter recommendation systems, the performance of user similarity measuring can be improved by filling the sparse marking matrix. Different filling method has different effect on similarity calculation result. To resolve this problem, this paper makes recommendation by using nearest neighbor algorithm on marking sets constructed by three kinds of filling methods separately, analyzes the measure performance optimized by two methods of traditional similarity measures and the similarity based on cloud model, and selects the most effective similarity measure optimization scheme for each filling method.

**【Key words】** collaborative filter; nearest neighbor; similarity degree; cloud model

## 1 概述

协同过滤是用于减少信息过载的常用技术，已成为个性化推荐系统的主要工具。最近邻协同过滤算法[1]是当前最成功的推荐技术之一。但随着推荐系统规模的扩大，用户评分数据出现极端稀疏性，导致该算法的推荐质量降低。

为解决数据稀疏性问题，一些学者提出了新的相似度计算模型，如文献[2]提出基于云模型的相似度计算方法。一些学者则采用对稀疏的用户-项矩阵进行填充的技术来提高相似度度量效果。最简单的填充办法是将用户对未评分项目的评分设为一个固定的缺省值，如设定为用户的平均评分，实验表明该方法可以有效提高协同过滤算法的推荐精度，因此，被许多简单推荐系统采用。另一种填充方法的处理过程如下：(1)采用预测评分的方式先估算出未评分项目的评分，将用户-项矩阵填充完整；(2)在得到的稠密矩阵上计算用户间的相似度，以最近邻算法进行推荐。例如，文献[3]提出一种基于项目评分预测的协同过滤推荐技术，通过估计用户评分来填充用户-项矩阵，减小数据稀疏性对计算结果的影响。文献[4]通过奇异值分解(Singular Value Decomposition, SVD)算法估计未评分项目的评分，并在稠密矩阵上计算用户间的相关相似度，采用最近邻算法求取实际未评分项目的预测值。

选取合适的相似度方法对提高推荐精度具有重要作用，因此，本文在 3 类填充后的评分数据集下对现有相似度度量方法进行了优化分析。

## 2 现有相似度度量方法

本文主要研究 4 种相似度：余弦相似度[2](Cos)，修正的余弦相似度[2](ACos)，相关相似度[2](Pearson)和基于云模型的相似度(Yun)。前 3 种相似度是传统相似度度量方法得到的，下文简要介绍基于云模型的相似度。云模型表达的概念的整体特性可以用期望 $Ex$、熵 $En$、超熵 $He$ 3 个特征来表示，记为 $\boldsymbol{C}(Ex, En, He)$，称为云的向量。在云模型中，云由多个云滴组成，每个用户的所有评分集合被视为一朵“云”，每个评分被视为一个“云滴”，可以通过逆向云算法[2]实现每朵云从定量值到云的特征向量的转换，2 朵云之间的相似度可以由云的特征向量的夹角余弦来表示。

基于云模型的相似度度量算法描述如下：

**输入** 用户 $i$ 的评分集合 $P_i=(x_1,x_2,\cdots,x_N)$，用户 $j$ 的评分集合 $P_j=(y_1,y_2,\cdots,y_M)$，其中，$N, M$ 分别为用户 $i$ 和用户 $j$ 评分过的项目个数。

**输出** 用户 $i$ 和用户 $j$ 的相似度 $YSim(i, j)$

(1)计算用户 $i$ 的评分矢量的样本均值 $\bar{X}=\frac{1}{N}\sum_{i=1}^{N}x_i$，一阶样本绝对中心矩 $\frac{1}{N}\sum_{i=1}^{N}\left|x_i-\bar{X}\right|$ 和样本方差 $S^2=\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{X})^2$。$Ex_i$ 的估计值为 $E\hat{x}=\bar{X}$，$He_i$ 的估计值为 $\hat{He}=\sqrt{\frac{\pi}{2}}\times\frac{1}{N}\sum_{i=1}^{N}\left|x_i-E\hat{x}\right|$，$En_i$ 的估计值为 $\hat{En}=\sqrt{s^2-\frac{1}{3}\hat{He}^2}$，则用户 $i$ 的云向量为 $\boldsymbol{C}_i=(Ex_i,En_i,He_i)$，用户 $j$ 的云向量为 $\boldsymbol{C}_j=(Ex_j,En_j,He_j)$。

(2)对任意 2 个用户 $i$ 和 $j$ 的相似度可以由 $\boldsymbol{C}_i$ 和 $\boldsymbol{C}_j$ 之间的余弦夹角来表示，即

**作者简介**：徐 翔(1984－)，男，硕士研究生，主研方向：电子商务个性化理论与方法；王煦法，教授、博士生导师

**收稿日期**：2009-10-25 **E-mail**：xuustc@gmail.com

$$YSim(i,j)=\cos(\boldsymbol{C}_i,\boldsymbol{C}_j)=\frac{\boldsymbol{C}_i\cdot\boldsymbol{C}_j}{\|\boldsymbol{C}_i\|\|\boldsymbol{C}_j\|}$$

## 3 2种相似度优化方法

### 3.1 基于用户评分项目并集的优化方法

对稀疏的评分矩阵采用填充技术后，用户间的所有评分项目均为共同评分项目，此时相关相似度和修正的余弦相似度是等同的。如果直接采用传统相似度方法计算最近邻，则用户的相似度由所有项目上的评分决定。而在实际系统中这是不现实的，因此，基于所有项目的相似度是不准确的。

文献[3]提出在预测评分矩阵上计算 2 个用户的相似度时，仅在用户实际评分项目的并集上进行，而不考虑他们均未评价的项目评分。

本文对云模型相似度采用文献[3]的思想进行修正。设用户 $i$ 和用户 $j$ 的评分项目的集合分别为 $I_i$ 和 $I_j$，他们评分的项目并集记作 $U_{ij}=I_i\cup I_j$，$i$ 和 $j$ 在 $U_{ij}$ 上的评分矢量分别记作 $\boldsymbol{UP}_i$ 和 $\boldsymbol{UP}_j$，且 $\boldsymbol{UP}_i=\{R'(i,t)\mid t\in U_{ij}\}$，$\boldsymbol{UP}_j=\{R'(j,t)\mid t\in U_{ij}\}$，$\boldsymbol{R}'$为填充后的用户-项矩阵，$R'(i,t)$表示用户 $i$ 对项目 $t$ 的评分。修正后，算法的输入变为 $UP_i$ 和 $UP_j$，其他部分与原算法相同。修正后基于云模型的相似度记作 UYun。同理，修正后的余弦相似度记作 UCos，修正后的相关相似度记作 UACos。

### 3.2 基于相关加权因子的优化方法

相关相似度虽然能反映用户间的相似程度，但不能显示相似度的可靠程度。例如，2 个用户在较少的项目上有相同评分不表示他们的兴趣相似，因为他们同时打过分的项目较少；如果 2 个用户共同评分的项目较多，则相关系数可以较准确地反映他们之间的相似关系。

一些推荐系统引入加权因子 $e$ 来提高相似度的可靠性，$e=Q/T$，其中，$Q$ 为 2 个用户共同评分的项目数；$T$ 为预先设定的阈值，与数据集大小相关。如果 $Q>T$，则加权因子为 1。经过修正后，与活动用户共同评价项目较少的用户对最终预测结果的贡献度有所降低。实验表明，加权因子可以提高相关相似度的可靠程度。基于上述思想，本文采用相关加权因子来修正其他相似度方法的度量结果，以提高协同过滤算法的推荐精度。

## 4 实验结果与分析

为分析以不同矩阵填充方式填充的用户评分数据对各相似度方法度量效果的影响，本文选取 3 种采用不同填充方式的数据集作为测试集，即极度稀疏数据集、缺省评分数据集和 SVD 预测评分数据集，并采用最近邻算法进行评分预测。实验以协同过滤算法的推荐精度来衡量各相似度方法度量效果的优劣。

### 4.1 数据集及度量标准

本文实验采用 Grouplens 工作组提供的公开数据集，它由 943 个用户的 10 万条值为 1~5 的评价数据组成。数据集中共有 1 682 个电影项目，每个用户至少对 20 个电影项目做出评价。实验选取其中 459 个用户的 2 万条评分数据作为初始测试集，稀疏等级为 97.4%。对稀疏评分矩阵分别进行 3 种填充操作，构建不同的测试集，即极度稀疏数据集(未评分项目的评分设置为 0)、缺省评分数据集(未评分项目的评分缺省设置为用户的平均评分)和 SVD 预测评分数据集(未评分项目的评分由 SVD 算法[5]估计)。

实验采用平均绝对偏差(Mean Absolute Error, MAE)来衡量各相似度方法的度量效果，即通过计算预测的用户评分与实际用户评分之间的偏差来衡量预测的准确性。$MAE$ 越小，相似度方法的度量效果越好。假设预测的用户评分集合为 $\{p1,p2,\cdots,pN\}$，对应的实际评分集合为 $\{q1,q2,\cdots,qN\}$，则

$$MAE=\frac{\sum_{i\in N}|p_i-q_i|}{N}$$

### 4.2 极度稀疏数据集

为评估各相似度方法的有效性，最近邻算法分别采用 Yun, Cos, ACos 和 Pearson 4 种相似度计算最近邻居集，并以 $MAE$ 来衡量相似度方法的度量效果。图 1 给出了极度稀疏数据集下各相似度方法的 $MAE$。结果表明，4 种方法的预测效果都不好，Pearson 的效果最差，ACos 次之，Cos 和 Yun 的效果较好，Cos 的效果最好。

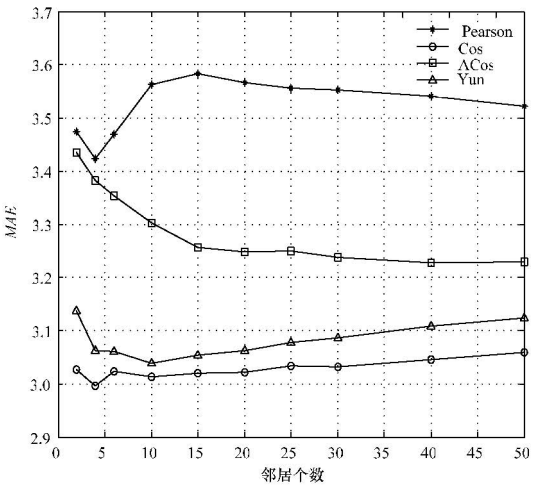


**图 1 极度稀疏数据集下各相似度方法的 *MAE***

对各相似度值的分布情况做了统计分析。Cos 有 63%的计算结果集中在(0.0, 0.2]的范围内，小于等于 0 的计算结果占 32%。Pearson 的结果在 0~1 范围内分布较均匀，但 60.6%的结果小于等于 0 或得不到结果。ACos 有 39%的结果集中在(0.0, 0.2]的范围内，59%的结果小于等于 0 或得不到结果。Yun 有 97%的结果分布在[0.9, 1.0]范围内，3%分布在[0.7, 0.9]范围内。

综上所述，从 0~1 范围的分布来看，Pearson 的分布结果最均匀，Yun 的分布情况最集中，Cos 和 ACos 其次。从结果小于等于 0 和得不到结果的总比重来看，Cos 和 Yun 的有效结果比例很大(分别为 68%和 100%)，Pearson 和 ACos 的无效结果均超过 60%。由各相似度结果的分布情况可知，Cos 和 Yun 的度量效果优于 ACos 和 Pearson。

### 4.3 缺省评分数据集

#### 4.3.1 基于用户评分项目并集的优化分析

在缺省评分数据集下，ACos 和 Pearson 等效，因此，实验只对 Cos, ACos 和 Yun 采用基于用户评分项目并集的修正，修正后的相似度分别为 UCos, UACos 和 UYun，并和原来的 3 种方法进行对比分析。如图 2 所示，UYun 的预测效果最好，最小 $MAE$ 为 0.846，Yun 次之。ACos 和 UAcos 方法在度量效果上几乎一致，UCos 的效果优于 Cos。可见，在缺省评分数据集上，云模型相似度比传统的余弦相似度和相关相似度的效果更好。基于用户评分项目并集的相似度修正方法提高了 Yun 和 Cos 的推荐精度，但没有改善 ACos 的效果。因此，在缺省评分数据集下，实验选定 UYun 相似度为度量效果最优的方法。

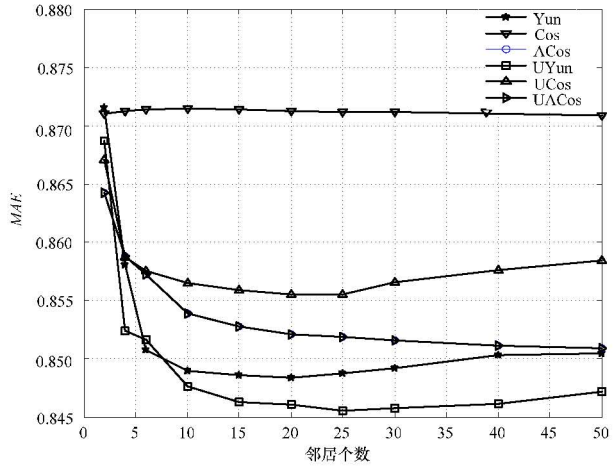


**图 2　缺省评分数据集下各相似度优化效果对比**

4.3.2　基于相关加权因子的优化分析

为验证相关加权因子能否对 UYun 相似度进一步优化，实验将 UYun 计算的相似度系数乘上加权因子，从而通过相关相似度的可靠性来修正 UYun 方法的相似度结果。

通过实验发现，加权因子中阈值 $T$ 的设定会影响相似度修正效果。本文将 $T$ 的取值设定为 10~50，以 10 为步长，分析不同取值对相似度修正结果的影响，结果如图 3 所示。UYun 表示未做加权因子修正的情况，UYT10 表示 UYun 经 $T$=10 的加权因子修正后的情况，以此类推。结果表明，$T$ 大于 40 后，推荐精度提升效果很小，因此，$T$ 最优设定为 40。

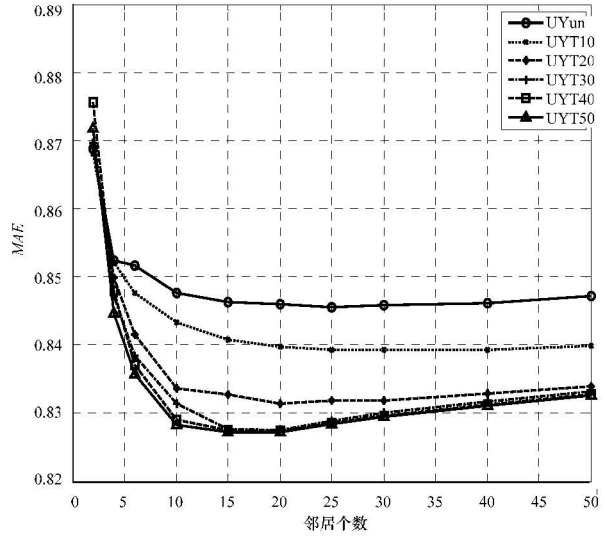


**图 3　UYun 以相关加权因子修正时不同阈值下的 *MAE***

### 4.4　SVD 预测评分数据集

4.4.1　SVD 保留维数的选取

在 SVD 算法预测评分前，需要先确定矩阵奇异值分解保留的维数 $k$。本文在初始测试集上进行多次实验，$k$ 从 1~25 递增，步长为 1，结果表明，当 $k$=8 时，算法的 *MAE* 最小为 0.83，预测精度最高。因此，SVD 保留的维数 $k$ 设定为 8。本文 SVD 的算法由 Matlab 工具实现。

4.4.2　基于用户评分项目并集的优化分析

SVD 保留维数确定后，分别以 Yun, Cos, ACos, UYun, UCos, UACos 6 种相似度计算最近邻居，做评分预测并与 SVD 算法的预测结果进行对比，如图 4 所示。只有 UACos 和 Yun 的预测精度超过了 SVD 算法。UACos 的最小 *MAE* 为 0.79，达到最优，Yun 方法次之。基于用户评分项目并集的相似度修正方法仅提高了 ACos 的效果，而降低了 Cos 和 Yun 的效果。因此，实验选取 UACos 方法为度量效果最优的相似度。

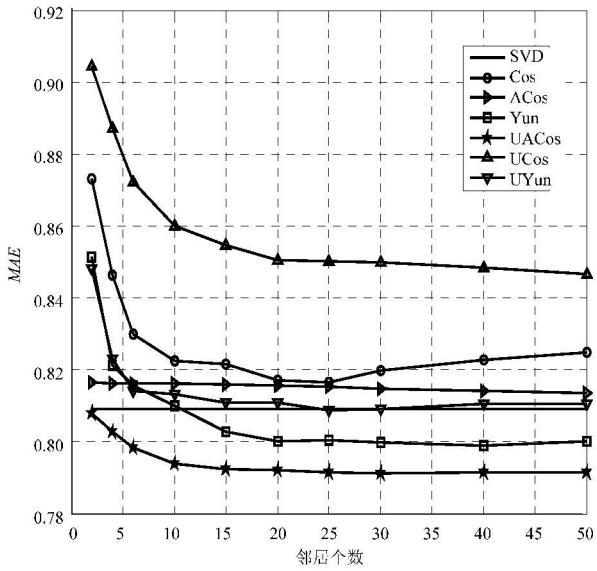


**图 4　SVD 预测评分数据集下各相似度优化效果对比**

4.4.3　基于相关加权因子的优化分析

为进一步提高算法精度，本文对 UACos 的结果以相关加权因子进行优化，并对阈值的选取进行实验，结果如图 5 所示，可以看出，随着阈值 $T$ 的增加，算法的推荐精度没有提高，反而下降。

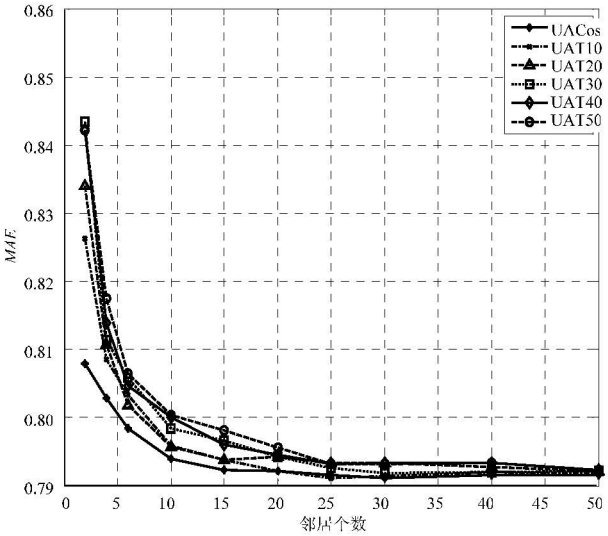


**图 5　UACos 以相关加权因子修正时不同阈值下的 *MAE***

### 4.5　实验结论

在极度稀疏数据集下，余弦相似度比其他相似度更简单且有效。在缺省评分数据集下，基于云模型的相似度经 2 种优化方法修正后(UYT40)的效果最优。在 SVD 预测评分数据集下，基于用户项目评分并集的修正 ACos 相似度(UACos)比其他相似度效果更好。对于后 2 种稠密数据集，Yun 和 ACos 及其优化后的度量效果较好，基于用户评分项目并集的优化方案显著提高了相似度的效果，但两者在基于相关加权因子的方法下的效果存在差异，其原因可能是 2 种数据集的填充评分的精确度不同。

(下转第 57 页)

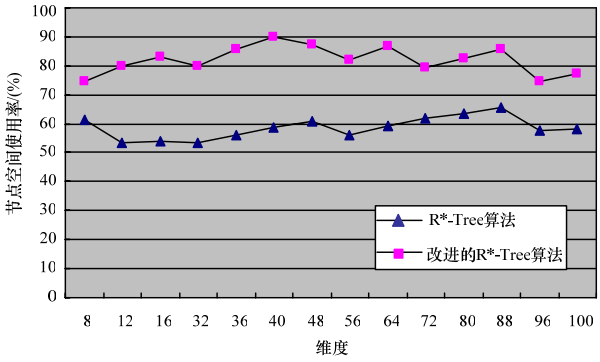


**图 4　节点空间利用率**

(2)2 种索引树的动态创建时间比较

创建索引时间平均缩短了 25.615 5%，如图 5 所示。

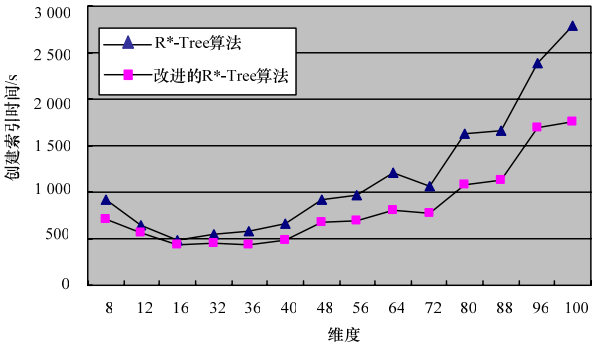


**图 5　创建索引时间**

### 3.4　*K*-最邻近查询方法的检索性能比较

采用*K*-最近邻查询[4](*K*=10)来比较 2 种索引的检索效率：

(1)2 种索引结构 I/O 的访问次数比较

磁盘的访问次数平均减少了 38.200 4%，如图 6 所示。

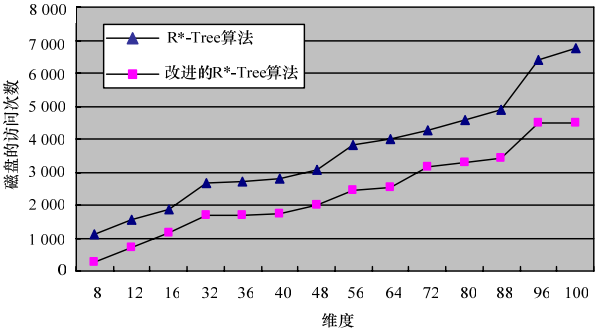


**图 6　磁盘的访问次数**

(2)2 种索引结构的查询时间比较

进行一次 *K*-最近邻查询(*K*=10)所需时间平均缩短了 22.405 8%，如图 7 所示。

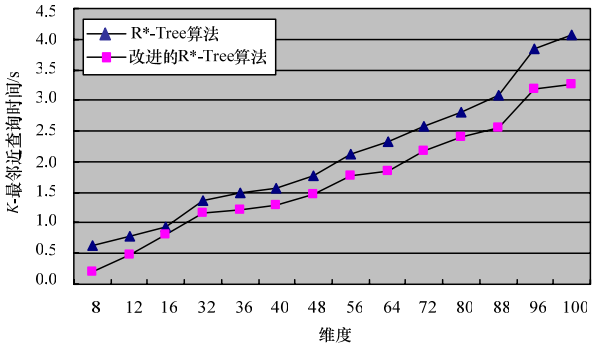


**图 7　*K*-最邻近查询时间**

## 4　结束语

本文改进 R*-Tree 的强制重插算法，当磁盘页溢出时，删除使 MBR 面积减少最多且使得 MBR 形状更“方”的项，再将其重新插入到索引中，使索引结构和索引性能得到较大优化。下一步工作是：进一步改进和优化该索引，并将此索引结构应用于基于图像语义特征的图像检索[5]研究中。

### 参考文献

[1] Guttman A. R-Trees: A Dynamic Index Structure for Spatial Searching[C]//Proc. of the International Conference on Management of Data. [S. l.]: ACM Press, 1984.

[2] Beckmann N. R*-Tree: An Efficient and Robust Access Method for Points and Rectangles[C]//Proc. of the 1990 ACM SIGMOD International Conference on Management of Data. Atlantic City, USA: ACM SIGMOD Press, 1990.

[3] Zhang Donghui, Xia Tian. A Novel Improvement to the R*-Tree Spatial Index Using Gain/loss Metrics[C]//Proc. of the 12th ACM International Workshop on Geographic Information Systems. Washington D. C., USA: [s. n.], 2004.

[4] Roussopoulos N, Kelley S, Vincent F. Nearest Neighbor Queries[C]// Proc. of the 1995 ACM SIGMOD International Conference on Management of Data. New York, USA: ACM SIGMOD Press,1995.

[5] 石跃祥, 朱东辉, 蔡自兴. 图像语义特征的抽取方法及其应用[J]. 计算机工程, 2007, 33(19): 177-179.

编辑　陆燕菲

(上接第 54 页)

## 5　结束语

本文在 3 类采用不同填充方式的数据集下，分析了传统相似度和云模型相似度经 2 种修正方法优化后的度量效果。通过实验选取各种情况下效果较好的相似度优化方案，结果表明，优化后的相似度均优于未经优化的相似度。

### 参考文献

[1] 郭艳红, 邓贵仕, 雒春雨. 基于信任因子的协同过滤推荐算法[J]. 计算机工程, 2008, 34(20): 1-3.

[2] 张光卫, 李德毅, 李　鹏. 基于云模型的协同过滤推荐算法[J]. 软件学报, 2007, 18(10): 2403-2411.

[3] 邓爱林, 朱扬勇, 施伯乐. 基于项目评分预测的协同过滤算法[J]. 软件学报, 2003, 14(9): 1621-1628.

[4] 孙小华, 陈　洪, 孔繁胜. 在协同过滤中结合奇异值分解与最近邻方法[J]. 计算机应用研究, 2006, 13(9): 206-208.

[5] Sarwar B M, Karypis G, Konstan J A. Application of Dimensionality Reduction in Recommender System——A Case Study[C]//Proc. of ACM WebKDD Workshop. [S. l.]: ACM Press, 2000.

编辑　陈　晖