Journal of Computer Applications
计算机应用,2012,32(3):658-660,668
ISSN 1001-9081
CODEN JYIIDU
2012-03-01
http://www.joca.cn

文章编号:1001-9081(2012)03-0658-03
doi:10.3724/SP.J.1087.2012.00658

基于项目属性和云填充的协同过滤推荐算法

孙金刚*,艾丽蓉
(西北工业大学 计算机学院,西安 710129)
(*通信作者电子邮箱 sjg_wo313@163.com)

摘　要:传统协同过滤推荐算法中经常因用户评分矩阵极端稀疏而导致相似性度量方法不准,推荐质量不高,针对这一问题,提出一种基于项目属性和云填充的协同过滤推荐算法。利用云模型对用户评分矩阵进行填充,在填充矩阵基础上,利用传统的相似性计算方法得到项目之间的评分相似性,同时结合项目属性,计算项目的属性相似性,通过加权因子得到项目的最终相似性,从而形成一种新的相似性度量方法。实验结果表明,提出的算法可有效解决传统方法中由于数据稀疏所导致的相似性度量不准确的问题,并显著地提高了算法的推荐精度。

关键词:协同过滤;稀疏数据;云填充;评分相似性;属性相似性;相似性度量
中图分类号:TP18;TP301.6　　**文献标志码**:A

Collaborative filtering recommendation algorithm based on item attribute and cloud model filling

SUN Jin-gang*, AI Li-rong
(*School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an Shaanxi* 710129, *China*)

Abstract: The user rating data in traditional collaborative filtering recommendation algorithm are extremely sparse, which results in bad similarity measurement and poor recommendation quality. In view of this problem, this paper presented an improved collaborative filtering algorithm, which was based on item attribute and cloud model filling. The algorithm proposed a new similarity measurement method, using the data filling based on cloud model and the similarity of the item's attributes. The new method computed the rating similarity by using the traditional similarity measurement on the basis of the filling matrix and computed the attributing similarity by using item's attributes, then got the last similarity by using weighting factor. The experimental results show that this method can efficiently solve the problem of similarity measurement inaccuracy caused by the extreme sparsity of user rating data, and provide better recommendation results than traditional collaborative filtering algorithms.

Key words: collaborative filtering; sparse data; cloud model filling; rating similarity; attributing similarity; similarity measure

0　引言

Internet 技术的应用普及和现代电子商务的迅猛发展使得互联网中的资源数量呈指数增长态势,从而出现了所谓的"信息爆炸"和"信息过载"现象,推荐系统[1]作为一种信息过滤的重要手段,是当前解决上述问题的有效方法之一。目前,几乎所有大型的电子商务系统,如淘宝、当当、Amazon、eBay 等都不同程度地使用了各种形式的推荐系统。

随着推荐系统规模的扩大,用户评分数据出现极端稀疏性[2],导致算法的推荐质量降低。为了解决稀疏性问题,一些学者提出了算法改进措施:文献[3]提出基于云模型的相似度计算方法,利用云模型在定性知识表示以及定性、定量知识转换时的桥梁作用,提出一种在知识层面比较用户相似度的方法。文献[4]提出一种基于项目评分预测的协同过滤推荐技术,通过估计用户评分的办法补充用户项目评分矩阵,减小数据稀疏性对计算结果的负面影响。文献[5]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。文献[6]提出了一种基于云模型数据填充的推荐方法,通过云模型预测未评价的项目,进而根据填充了的用户项目评分矩阵计算用户相似性。然而,以上算法均没有考虑项目中属性之间的关联,忽略了项目本身的属性相似性,在一定程度上影响了算法的效能。

本文提出一种基于项目属性和云模型数据填充方法,利用项目本身属性计算属性相似性,利用云模型数据填充技术对稀疏矩阵进行填充,在稠密矩阵上计算项目评分相似性,通过动态加权得到新的项目相似性,最后在推荐过程中结合基于项目推荐算法生成推荐。

1　传统的基于项目的协同过滤算法及分析

传统的基于项目的协同过滤推荐系统[7]利用传统的相似性计算方法计算所有项目与目标项目的相似性,取前 k 个相似性最大的作为目标项目的最近邻居集,由于当前用户对最近邻居的评分与对目标项的评分比较类似,所以可以根据当前用户对最近邻居的评分预测当前用户对目标项的评分,然后选择预测评分最高的前若干项作为推荐结果反馈给用户。

1.1　数据表示

推荐系统中存储的用户评分数据中一般包含用户 id,项目 id 和用户对项目的评分等信息。假定有 m 个用户和 n 个项

收稿日期:2011-08-19;**修回日期**:2011-11-26。
作者简介:孙金刚(1981-),男,河北唐山人,硕士研究生,主要研究方向:智能推荐;　艾丽蓉(1970-),女,陕西延安人,副教授,博士,主要研究方向:智能信息处理。

目,分别表示为用户集合 $U=\{u_1,u_2,\cdots,u_m\}$,项目集合 $I=\{i_1,i_2,\cdots,i_n\}$。用户评分数据可以采用一个 $m\times n$ 阶的用户项目评分矩阵 $\boldsymbol{R}_{m,n}$ 来表示,矩阵的每个元素就是用户对项目的评分,反映了用户对项目感兴趣的程度。评分可以用 1/0 表示喜欢 / 不喜欢;也可以用不同的数字表示评分的级别,数字越大表明级别越高,说明用户越喜欢(本文即用这种评分)。

1.2　相似性计算

传统的相似性度量[1,8]方法主要有余弦相似性、相关相似性和修正的余弦相似性 3 种。

1)余弦相似性。两个项目被看作 m 维项目空间上的向量,项目间的相似性通过向量间的余弦夹角来度量。

$$sim(i,j)=\cos(\vec{i},\vec{j})=\frac{\vec{i}\cdot\vec{j}}{\|\vec{i}\|\times\|\vec{j}\|} \tag{1}$$

其中"·"表示两个向量之间的点积。式中分子为两个项目评分向量的内积,分母为两个项目向量模的乘积,夹角越小,相似度越高。

2)相关相似性。对于基于项目的协同过滤算法,相关相似性是通过计算两个项目之间共同评分项的距离来实现的。

$$sim(i,j)=\frac{\sum_{u\in U(i,j)}(R_{u,i}-\overline{R_i})(R_{u,j}-\overline{R_j})}{\sqrt{\sum_{u\in U(i,j)}(R_{u,i}-\overline{R_i})^2}\sqrt{\sum_{u\in U(i,j)}(R_{u,j}-\overline{R_j})^2}} \tag{2}$$

其中:$U(i,j)$ 表示对项目 i 和项目 j 都评过分的用户集合,$R_{u,i}$、$R_{u,j}$ 表示用户 u 对项目 i 和项目 j 的评分,$\overline{R_i}$ 和 $\overline{R_j}$ 分别表示对项目 i 和项目 j 的平均评分。

3)修正的余弦相似性。由于在余弦相似性度量方法中没有考虑不同用户的评分尺度问题,修正的余弦相似性度量方法通过减去用户对所有项目的平均评分来改善上述缺陷。

$$sim(i,j)=\frac{\sum_{u\in U(i,j)}(R_{u,i}-\overline{R_i})(R_{u,j}-\overline{R_j})}{\sqrt{\sum_{u\in U(i)}(R_{u,i}-\overline{R_i})^2}\sqrt{\sum_{u\in U(j)}(R_{u,j}-\overline{R_j})^2}} \tag{3}$$

其中:$U(i,j)$ 表示对项目 i 和项目 j 都有评分的用户集合,$U(i)$ 和 $U(j)$ 分别表示对项目 i 和项目 j 评分过的用户集合。

1.3　生成推荐

在利用 1.2 节中任一公式计算目标项目与其他项目相似性,得到目标项目最近邻居集的基础上,可按式(4)预测出目标用户对未评分项目的评分。

$$P_{u,i}=\overline{R_i}+\frac{\sum_{j\in S(i)}sim(i,j)\times(R_{u,j}-\overline{R_j})}{\sum_{j\in S(i)}(|sim(i,j)|)} \tag{4}$$

其中:$P_{u,i}$ 表示目标用户 u 对未评分项目 i 的预测评分;$\overline{R_i}$ 表示目标项目 i 获得的平均评分;$\overline{R_j}$ 表示邻近项目 j 的平均评分;$S(i)$ 表示目标项目 i 的邻近项目集合,$|S(i)|=k$。通过上述方法预测用户对所有未评分项的评分,然后选择预测评分最高的前若干项作为推荐结果反馈给当前用户。

1.4　算法分析

1)用户评分矩阵的极端稀疏性导致相似度量不够准确,严重影响了推荐精度。

2)没有考虑到项目本身的属性相似性,利用用户评分数据计算的相似性可能不准确。如表 1 所示,I_1 到 I_6 代表 6 部电影,其中 I_1、I_3、I_6 为生活片,I_2、I_4、I_5 为动作片,现要预测用户 u_7 对 I_6 的评分。根据式(2)、(4)可得 I_6 的邻居项为 I_2、I_4、I_5,u_7 对 I_6 的预测评分为 1,即不喜欢 I_6,但分析 u_7 对其他电影的评分可以看到,u_7 对生活片的评分均为 5,对动作片的评分均为 1,由此可判断 u_7 喜欢生活片,不喜欢动作片,故对 I_6 应该打高分,但通过公式得到的计算结果却与实际不符。

表 1　用户评分实例

预测用户	I_1	I_2	I_3	I_4	I_5	I_6
u_1	3	3	3	3	2	3
u_2	1	2	2	3	3	3
u_3	4	5	4	4	5	5
u_4	5	5	3	3	5	5
u_5	2	1	1	1	2	1
u_6	1	4	1	1	1	1
u_7	5	1	5	1	1	?

2　改进的算法及其分析

2.1　云模型

云模型[9-10]表达的概念的整体特征用期望 Ex、熵 En、超熵 He 这 3 个数字特征来表示,记作 $\boldsymbol{C}(Ex,En,He)$,称为云的特征向量。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实现每朵云从定量值到云的特征向量的转换,计算如下:

$$Ex:\overline{Ex}=\overline{X} \tag{5}$$

$$He:\overline{He}=\sqrt{\frac{\pi}{2}}\times\frac{1}{N}\sum_{i=1}^{N}|x_i-\overline{Ex}| \tag{6}$$

$$En:\overline{En}=\sqrt{S^2-\frac{1}{3}\overline{He}^2} \tag{7}$$

其中:$\overline{X}$ 为样本均值,S^2 为样本方差。

$$\overline{X}=\frac{1}{N}\sum_{i=1}^{N}x_i$$

$$S^2=\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\overline{X})^2$$

两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,计算如下:

$$sim_Y(i,j)=\cos(\overrightarrow{C_i},\overrightarrow{C_j})=\frac{\overrightarrow{C_i}\cdot\overrightarrow{C_j}}{\|\overrightarrow{C_i}\|\times\|\overrightarrow{C_j}\|} \tag{8}$$

2.2　基于云模型填充的协同过滤推荐算法

基于云模型填充的协同过滤推荐算法[6-11]的基本思想是利用逆向云算法计算项目之间的相似性,根据式(4)对用户评分矩阵进行填充,在填充矩阵的基础上,根据式(2)、(4)生成推荐。

该算法通过填充的方法降低了用户评分矩阵的稀疏性,提高了推荐精度,但是并未考虑到项目本身的属性相似性。

2.3　项目的属性相似性

一般来说,推荐系统至少存在 3 个基本数据表,一个用来记录用户信息,一个用来记录项目信息,还有一个记录用户评分信息。传统的协同过滤算法是通过评分信息表计算项目相似性,而项目的属性相似性则是利用项目信息表(见表 2)进行计算的。

表 2　项目信息表

项目	$Attr_1$	$Attr_2$	…	$Attr_t$
$Item_1$	1	0	…	1
$Item_2$	0	1	…	1
⋮	⋮	⋮		⋮
$Item_n$	1	1	…	0

设项目 i 和项目 j 所拥有的属性分别表示为集合 U_i 和集合 U_j，则 i 和 j 的属性相似性计算如下：

$$sim_{attr}(i,j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|} \tag{9}$$

其中：$|U_i \cap U_j|$ 表示项目 i 和 j 所拥有属性集合的交集的元素个数，$|U_i \cup U_j|$ 表示项目 i 和 j 所拥有属性集合的并集的元素个数[12]。

2.4 改进的基于项目属性和云填充的协同过滤算法

改进的基于项目属性和云填充的协同过滤算法的基本思想是：利用用户评分矩阵和逆向云算法计算项目 i 和项目 j 的相似性，此时将每个项目所获得的所有评分集合视为一朵“云”，每个评分被视为一个“云滴”。得到项目 i 和 j 的相似性后，利用式(4) 对稀疏的用户评分矩阵进行填充，采用对项目评分项并集填充的方式[4]，即对于项目 i 和 j，对项目 i 评过分的用户集合为 U_i，对项目 j 评过分的用户集合为 U_j，只填充 $U_{ij} = U_i \cup U_j$ 中没有对项目 i 或 j 评分的用户的评分，而不考虑均未对项目 i 和 j 评分的用户。在填充矩阵的基础上利用式(2) 计算项目 i 和 j 的评分相似性 $sim_{rate}(i,j)$，同时根据式(9) 计算项目 i 和 j 的属性相似性 $sim_{attr}(i,j)$，采用式(10) 计算项目 i 和 j 的最终相似性 $sim(i,j)$[13]，最后再次利用式(4) 产生推荐。

$$sim(i,j) = \lambda sim_{attr}(i,j) + (1-\lambda) sim_{rate}(i,j) \tag{10}$$

其中：λ 为加权因子，可动态调整。

具体步骤如下：

输入　用户评分矩阵 $\boldsymbol{R}_{m,n}$，最近邻项目数 k，加权因子 λ。

输出　用户的 Top-N 推荐集。

步骤如下：

1) 利用逆向云算法即式(5) ~ (7) 计算每个项目的特征向量，并利用式(8) 计算项目之间的相似性；

2) 利用式(4)，采用对项目评分项并集的方式对用户评分矩阵进行填充；

3) 对于填充矩阵，利用式(2) 计算项目 i 和 j 的评分相似性 $sim_{rate}(i,j)$；

4) 利用式(9) 计算项目 i 和 j 的属性相似性 $sim_{attr}(i,j)$；

5) 利用式(10) 计算项目 i 和 j 的最终相似性 $sim(i,j)$；

6) 选取 k 个与目标项目相似性最大的项目作为目标项目的最近邻居集；

7) 利用式(4) 得到目标用户对初始矩阵中未评分项目的预测评分，产生 Top-N 推荐。

算法优势分析：首先，采用填充技术对稀疏矩阵进行填充，在填充矩阵基础上计算项目的评分相似性，大大降低了原有稀疏矩阵的稀疏性，提高了评分相似性的计算精度；其次，采用云模型而非其他填充模型，可以在用户评分整体层面上粗粒度地考虑项目的相似性以及对象类的整体信息，克服了传统的基于向量的相似度计算方式严格匹配对象属性的不足。再次，考虑了项目本身所具有的属性相似性，通过引入加权因子 λ，将其与项目的评分相似性结合，进一步提高了相似性精度。同时，对于用户从未评过分的新项目，在无法计算其与其他项目的评分相似性的情况下，将加权因子 λ 调整为 1，即通过属性相似性来计算相似性，在一定程度上解决了冷启动问题。

3 实验结果及其分析

实验平台使用 Intel Core2 Duo CPU E8400 @3.00 GHz，1.96 GB内存，操作系统为 Windows XP，算法使用C++编写。

3.1 数据集

本文实验采用 Grouplens(http://movielens.umn.edu) 工作组提供的公开数据集，它由 943 个用户对 1 682 个电影项目的 1×10^5 条值为 1 ~ 5 的评价数据组成，用户评分数据集的稀疏性为 1 - 100 000/(943 * 1 682) = 0.936 953。把记录按照 80% 和 20% 的比例划分为训练集和测试集。

3.2 度量标准

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[14-15]。*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 $\{p_1,p_2,\cdots,p_N\}$，对应的实际评分集合为 $\{q_1,q_2,\cdots,q_N\}$，则

$$MAE = \frac{\sum_{i=1}^{N}|p_i - q_i|}{N} \tag{11}$$

3.3 实验结果

实验中，将最近邻近集取为 50，通过改变 λ 取值来影响 *MAE* 的值，具体过程为：将 λ 值代入式(10)，得到相应的最终相似性 $sim(i,j)$，选取 50 个与目标项目相似性最大的项目作为目标项目的最近邻居集，将 $sim(i,j)$ 值代入式(4)，得到对目标用户的预测评分值，再将预测值代入式(11)，得到 *MAE* 值。实验结果如表 3 所示，当最近邻近集 k 取 50 时，λ = 0.15 时对 *MAE* 的贡献最大，故实验中将 λ 取为 0.15。

表 3　λ 取值对 *MAE* 的影响

λ	*MAE*	λ	*MAE*
0.05	0.824	0.20	0.856
0.10	0.817	0.25	0.932
0.15	0.811	0.30	1.125

当 λ 取值确定后，将改进的算法与传统的基于项目的协同过滤推荐算法、基于云模型的协同过滤推荐算法和利用云模型填充的协同过滤算法进行比较，如图 1 所示。图中 Item 表示传统的基于项目的协同过滤推荐算法，Cloud 表示基于云模型的协同过滤推荐算法，Cloud Filling 表示利用云模型填充的协同过滤推荐算法，Attribute and Cloud Filling 表示改进的基于项目属性和云填充的协同过滤推荐算法，最近邻居从 5 个递增到 50 个。通过比较可以看出，改进的基于项目属性和云填充的协同过滤算法比上述 3 种算法具有更好的性能。

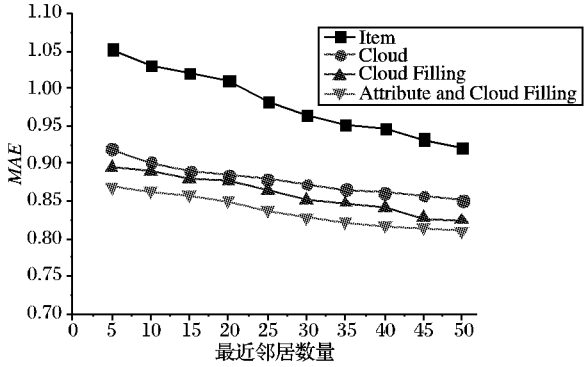

图 1　不同算法 *MAE* 值随最近邻居数变化对比

4 结语

本文分析了传统的基于项目协同过滤算法及其存在的问题，提出了一种改进的基于项目属性和云填充的协同过滤推

(下转第 668 页)

能的影响。标签 ID 越长，可利用的碰撞比特信息就越多，当标签 ID 长度为 128 时，算法会根据节点内标签的数量自适应地选择二叉树、四叉树和八叉树搜索进行搜索。而标签 ID 长度为 32 或 64 时，根据式(9)，$1 \leqslant D \leqslant 2$，算法只能选择二叉树和四叉树，因此算法性能在标签长度为 128 位时，要优于 32 或 64 位，而在后两者情况下，算法的性能相近。

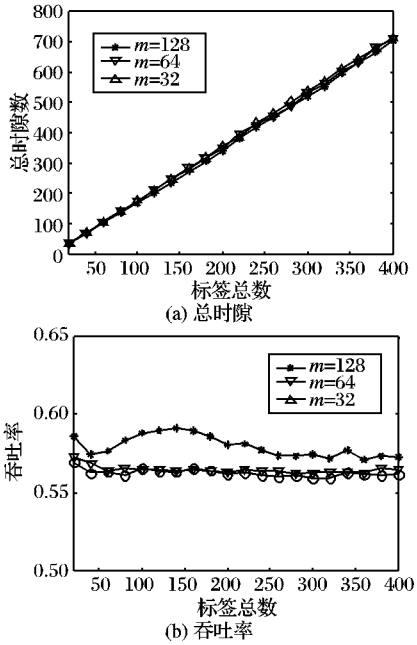

图 4 标签 ID 长度对算法性能的影响

值得注意的是，在新算法中无论是启发式函数的计算还是搜索叉数的确定都是在读写器端实现的，不需要增加标签的硬件成本，因此新算法具有较高的实用性。

5 结语

本文提出了一种基于启发式函数的自适应多叉树防碰撞算法。该算法利用曼彻斯特编码可以识别碰撞位的特性，通过定义和计算启发式函数，以估计节点内待识别标签的数量，可在不同搜索节点和深度，自适应地调整搜索叉数，从而有效地提高了算法的搜索效率。新算法克服了传统防碰撞算法的缺点，尤其在待识别标签数量较多场合，可有效地减少搜索和识别时间，提高 RFID 系统的吞吐率。

参考文献：

[1] FINKENZELLER K. RFID Handbook: fundamentals and applications in contactless smart cards and identification[M]. Hoboken: John Wiley & Sons, 2003.

[2] HWANG T-W, LEE B-G, KIM Y-S. Improved anti-collision scheme for high speed identification in RFID system [C]// Proceedings of First International Conference on Innovative Computing, Information and Control. Piscataway, NJ: IEEE Press, 2006: 449-452.

[3] KIM J G. A divide-and-conquer technique for throughput enhancement of RFID anti-collision protocol [J]. IEEE Communications Letters, 2008, 12(6): 474-476.

[4] EOM J B, LEE T J, RIETMAN R. An efficient framed-slotted ALOHA algorithm with pilot frame and binary selection for anti-collision of RFID tags [J]. IEEE Communications Letters, 2008, 12(11): 861-863.

[5] JIHOON M, WONJUN L, SRIVASTAVA J. Adaptive binary splitting for efficient RFID tag anti-collision [J]. IEEE Communications Letters, 2006, 10(3): 144-146.

[6] LAI Y-C, LIN C-C. A pair-resolution blocking algorithm on adaptive binary splitting for RFID tag identification [J]. IEEE Communications Letters, 2008, 12(6): 432-434.

[7] CHOI J H, LEE D, LEE H. Query tree-based reservation for efficient RFID tag anti-collision [J]. IEEE Communications Letters, 2007, 11(1): 85-87.

[8] RYU J, LEE H, SEOK Y. A hybrid query tree protocol for tag collision arbitration in RFID systems [C]// ICC'07: IEEE International Conference on Communications. Piscataway, NJ: IEEE Press, 2007: 5981-5986.

[9] 丁治国，朱学永，郭立，等. 自适应多叉树防碰撞算法研究[J]. 自动化学报，2010，36(2)：237-241.

(上接第 660 页)

荐算法，最后通过实验验证了该算法能够有效地提高推荐精度。下一步将考虑用户评分数据对云模型推荐精度的影响，如评分用户多少、两个用户共同评分的多少对云模型相似度的影响。此外，考虑如何综合利用各种推荐系统，解决推荐系统中的冷启动问题。

参考文献：

[1] 刘建国，周涛，汪秉宏. 个性化推荐系统的研究进展[J]. 自然科学进展，2009，19(1)：1-15.

[2] 孙小华. 协同过滤系统的稀疏性与冷启动问题研究[D]. 杭州：浙江大学，2005.

[3] 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.

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

[5] 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，2006，23(9)：206-208.

[6] 余志虎，戚玉峰. 一种基于云模型数据填充的算法[J]. 计算机技术与发展，2010，20(12)：35-37.

[7] SARWAR B, KARYPIS G, KONSTAN J, *et al*. Item-based collaborative filtering recommendation algorithms [C]// Proceedings of the 10th International World Wide Web Conference. New York: ACM Press, 2001: 285-295.

[8] 白丽君，张永奎，陈鑫卿. 协作过滤研究概述[J]. 电脑开发与应用，2002，15(11)：2-3.

[9] 李德毅，刘常昱. 论正态云模型的普适性[J]. 中国工程科学，2004，6(8)：28-34.

[10] 李德毅，刘常昱，杜鹢，等. 不确定性人工智能[J]. 软件学报，2004，15(11)：1583-1594.

[11] 张新香，刘腾红. 利用云模型改进基于项目的协同过滤推荐算法[J]. 图书情报工作，2009，53(1)：117-120.

[12] 汪静，印鉴. 一种优化的 Item-based 协同过滤推荐算法[J]. 小型微型计算机系统，2010，31(12)：2338-2342.

[13] 徐翔，王煦法. 协同过滤算法中的相似度优化方法[J]. 计算机工程，2010，36(6)：52-57.

[14] BREESE J, HECKERMAN D, KADIE C. Empirical analysis of predictive algorithms for collaborative filtering [C]// Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. Madison: Morgan Kaufmann, 1998: 43-52.

[15] KONSTAN J, MILLER B, MALTZ D, *et al*. Grouplens: Applying collaborative filtering to usenet news [J]. Communications of the ACM, 1997, 40(3): 77-87.