

一种基于和声搜索的协同过滤算法研究*

王华秋

(重庆理工大学计算机学院 重庆 400054)

【摘要】改进传统的相似度计算方法,为寻找最优的相似度函数,采用参数优化的和声搜索算法来寻找相似度函数的最优权值向量。为提高推荐速度,得到最优的相似度函数后,对于用户的推荐计算不再采用和声搜索算法。实验表明,和传统算法相比,该算法能提高预测精度和覆盖率,有更好的推荐效果,并能够更快地获得目标用户的最邻近用户,加快推荐的速度。

【关键词】协同过滤 相似度函数 权值向量 和声搜索算法

【分类号】TP311 TP391

Research of a Collaborative Filtering Algorithm Based on Harmony Search

Wang Huaqiu

(School of Computer Science, Chongqing University of Technology, Chongqing 400054, China)

【Abstract】The traditional similarity algorithm of collaborative filtering is modified in this paper. In order to find an optimal similarity function, the paper presents harmony search algorithm with parameters optimization to find the optimal weights vector of similarity function. To improve the speed of recommendation, harmony search algorithm is no longer used for the calculation of the recommendation after finding the optimal similarity function. The validation experiments show that the proposed algorithm improves prediction accuracy and coverage so as to provide better recommendation. And the proposed algorithm can more quickly obtain the nearest neighbor users of the target user, which can accelerate the recommended speed.

【Keywords】Collaborative filtering Similarity function Weights vector Harmony search algorithm

1 引言

个性化推荐技术的基本原理是期望用户组和目标用户相似,这样就能比较正确地预测目标用户对没有进行评价的产品的兴趣度^[1]。近年来,个性化推荐技术越来越重要,已经应用在很多领域中,包括数字图书馆^[2]。

个性化推荐技术的核心技术是过滤算法。协同过滤(Collaborative Filtering, CF)是一种常用的过滤算法,其原理是:首先搜索与目标用户习惯最接近的用户群体,然后将大多数相似用户评价最高的项推荐给目标用户^[3]。能否及时准确地找到目标用户的最近邻居关系到整个系统的推荐质量,如何进行准确并快速地计算用户相似性成为提高推荐准确性和高效性的关键。常用的相似性度量方法有余弦相似性^[4](简称 COS)、Pearson 相关系数^[5](简称 COR)和约束 Pearson 相关系数^[6](简称 CCOR)。

与传统的推荐技术相比,协同过滤能够过滤掉多数难以自动进行内容分析的信息,还可以共享他人的经验,

收稿日期:2012-10-28

收修稿日期:2012-11-28

* 本文系教育部人文社会科学研究青年基金项目“虚拟专用网环境下图书馆服务多引擎专家系统的研制”(项目编号:10YJC870037)的研究成果之一。

具备推荐新信息的能力。但是由于需要利用用户对项目的评分数据,协同过滤存在以下需要解决的问题^[3]:

(1)稀疏性:在许多推荐系统中,用户-项目评估矩阵相当稀疏,一般用户评价信息所涉及到的商品只能占总数的1%~2%。

(2)精确性:在用户评分数据极端稀疏的情况下,传统相似性度量方法存在一定的弊端,使得计算得到的目标用户的最近邻居不准确,推荐系统的推荐质量急剧下降。

(3)效率性:随着用户数据量的增大,传统算法将遭遇严重的效率问题,造成推荐时间过长或系统瘫痪。

为了产生精确的推荐,并满足推荐系统的实时性要求,许多学者将多种人工智能算法应用于推荐系统中^[7-9]。一种常用的用于提高推荐系统性能的技术就是将用户进行聚类,这样可以得到相似的用户,然后,协同过滤技术就可以用于每一类相似用户中,这样可以减少计算时间,文献[7]提出了一种遗传聚类算法用于协同过滤的推荐系统中。文献[8]提出了基于聚类免疫的协同过滤推荐算法。文献[9]利用蚁群算法实现用户聚类,以提高协同过滤推荐系统的最近邻查询速度。

本文研究了一种新的智能优化算法——和声搜索算法,该算法与遗传算法、粒子群算法、蚁群算法一样,都是一种启发式搜索算法,已经被应用于许多方面^[10-12]。由于其调音概率和随机带宽这两个参数设置随机性较大,缺乏方向性,造成和声搜索算法在一些情况下不能很好地搜索到精度较高的解,而且容易陷入局部最优,针对这些问题,本文提出了和声搜索的参数优化方法,并将参数优化的和声搜索算法应用到协同过滤中来,提出了一种基于和声搜索的协同过滤推荐技术,将和声搜索与协同过滤算法的优点结合,对协同过滤算法的相似度函数进行改进,以解决稀疏性、精确性和效率性等问题,可以产生精确的个性化推荐结果。

2 改进的协同过滤相似度函数

假设有一组用户 $U = \{1, \dots, U\}$, 以及一组项目 $I = \{1, \dots, I\}$ 。用户的评价范围是 m 到 M , m 表示用户对项目不满意, M 表示用户对项目完全满意。

对于 MovieLens 数据库 (<http://movielens.umn.edu>), 可以设 $m = 1, M = 5$ 。

用户一般只会对感兴趣的项目进行评分,没有评分的项目本文用 null 表示,如下所示:

$$r_x = (r_x^{(1)}, r_x^{(2)}, \dots, \text{null}, \dots, r_x^{(M)})$$

由于用户参与评分的项目并不多,这就造成了用户-项目矩阵的稀疏性。传统的相似度计算方法,如 $\text{COS}^{[4]}$, $\text{COR}^{[5]}$, $\text{CCOR}^{[6]}$ 等,都无法妥善处理矩阵中的 null 项,从而造成误差较大,基于此本文改进了传统的相似度计算方法。

对于任意两个用户 u 和 n , 假定总有都进行了评价的项目,这些共同项目可以计算出 x 和 y 的关系,在这些项目中, x 和 y 评价的分数之差的绝对值在 0 和 $M - m$ 之间,为了将这些评分差值归一化,用这些差异评分值除以 u 和 n 共同评价的项目总和,这样每一项的值就不会超过 1,如下所示:

$$r_{u,n} = (r_{u,n}^{(0)}, r_{u,n}^{(1)}, \dots, r_{u,n}^{(M-m)})$$

因此可以得到 u 和 n 对所有评价项目的差别如下:

$$r_{u,n} = (r_{u,n}^{(0)}, r_{u,n}^{(1)}, \dots, r_{u,n}^{(4)}) = (\frac{1}{5}, \frac{1}{5}, \frac{2}{5}, \frac{1}{5}, 0)$$

这样做的优点在于,无论 u 和 n 的评分项目有多少,都可以产生两者之间的关系向量 $r_{u,n}$,进而计算两者的相似度。

为计算相似度,为每个评分差异项 $r_{u,n}$ 进行加权,权值向量 w 和 $r_{u,n}$ 对应 $w = (w^{(0)}, w^{(1)}, \dots, w^{(M-m)})$,对于 MovieLens 数据库,按照如下原则设定:

(1)对于 $r_{u,n}^{(0)}$ 值,这说明 u 和 n 对于某些项目的评分分数完全相同,可以认为 u 和 n 对于这些项目完全接近,因而将 $r_{u,n}^{(0)}$ 对应的 $w^{(0)}$ 值可以设为 1。

(2)对于 $r_{u,n}^{(1)}$ 值,这说明 u 和 n 对于某些项目的评分分数仅仅差 1 分,可以认为 u 和 n 对于这些项目比较接近,因而将 $r_{u,n}^{(1)}$ 对应的 $w^{(1)}$ 值可以设为 0.5。

(3)对于 $r_{u,n}^{(2)}$ 值,这说明 u 和 n 对于某些项目的评分分数相差 2 分,在评分差别为 4 的评分等级中,可以认为 u 和 n 对于这些项目接近程度一般,因而将 $r_{u,n}^{(2)}$ 对应的 $w^{(2)}$ 值可以设为 0。

(4)对于 $r_{u,n}^{(3)}$ 值,这说明 u 和 n 对于某些项目的评分分数相差 3 分,可以认为 u 和 n 对于这些项目差别较大,因而将 $r_{u,n}^{(3)}$ 对应的 $w^{(3)}$ 值可以设为 -0.5。

(5)对于 $r_{u,n}^{(4)}$ 值,这说明 u 和 n 对于某些项目的评分分数相差 4 分,在评分差别为 4 的评分等级中,可以

认为 u 和 n 对于这些项目的评价完全不同,因而将 $r_{u,n}^{(4)}$ 对应的 $w^{(4)}$ 值可以设为 -1 。

为 $r_{u,n}$ 得到如下权值向量:

$$w = (w^{(0)}, w^{(1)}, \dots, w^{(M-m)}) = (1, 0.5, 0, -0.5, -1)$$

本文定义 u 和 n 的相似度函数:

$$\text{sim}_w(u, n) = \frac{1}{M-m+1} \sum_{i=0}^{M-m} w^{(i)} r_{x,y}^{(i)} \quad (1)$$

本文采用共同评分项的差异值 $r_{x,y}$ 解决了评分矩阵的稀疏性问题,但是,新的问题又出现了,从上面的推导中可以看出, $r_{x,y}$ 是一个客观的值,但是其权值向量 w 却是一个主观设定、由人为确定的相关权值系数,但是这样的主观设定并不能保证获得一个最佳的权重向量。如何寻找一个最优的权值向量以获得最合适的相似度函数,这一问题关系到协同过滤推荐系统是否有效。不妨假设,在所有可能的权值向量 w 的取值中,总有一个向量能为本文算法提供一个具有最小平均绝对误差 (Mean Absolute Error, MAE) 的一组向量。这就需要通过不断训练和测试过程而获得最优推荐系统,为获得这个问题的最优解,本文采用和声搜索算法。

3 和声搜索算法的研究与改进

为了找到最优的相似度函数,本文采用和声搜索算法来寻找相似度函数最优的 w 权值。本文采用监督学习策略,其适应度函数是推荐系统的平均绝对误差 MAE。权值向量的不同取值构成和声搜索算法的和声库,库中每一组和声向量代表一组可能的权值,本算法将用这样的值计算推荐系统的误差。当算法运行的时候,连续的和声库向量迭代将改善推荐系统的 MAE。当 MAE 低于预期的阈值时和声库向量的迭代终止。

本文只使用推荐系统的一部分数据,以获得最优的相似度权值,从而获得最优的相似度函数,之后本文将利用这个相似度函数来计算测试用户和项目的相似性,即是说,这些测试用户和项目的相关计算不再用到和声搜索算法,从而提高了计算速度。

3.1 和声库向量表示

和声库向量一般都用实数表示^[13]。本文的 w 向量的任意元素 $w^{(i)}$ 用小数表示,因此,用下面的方法表示 $w = (w^{(0)}, w^{(1)}, \dots, w^{(M-m)})$:

$$HM = \left[\begin{array}{cccccc} 0.82, 0.31, 0.15, -0.45, -0.75 \\ \dots \\ \dots \\ 0.72, 0.46, 0.23, -0.35, -0.87 \end{array} \right] \quad \text{HMS}$$

其中,和声库 HM 的行数为 HMS,本文设为 50,代表 50 种不同的权值向量组;列数为 d,本文设为 5,代表每组权值向量的个数。每一行代表一组权值向量,这些向量可以直接用于相似度函数的计算,而无需任何转换,这比遗传算法更加方便。

3.2 初始化和声库

既然 w 的值代表用户之间相似度函数的权重,那么可以认为在最优的相似度函数中,按照不同的权值代表的不同的用户之间的相关性,把 $w^{(0)}, w^{(1)}, \dots, w^{(M-m)}$ 依次放在正相关到负相关逐渐过渡的范围中。据此,得到如下的 w 初始化范围分布:

$$w_0 \in [0.6, 1], w_1 \in [0.2, 0.6], w_2 \in [-0.2, 0.2],$$

$$w_3 \in [-0.6, -0.2], w_4 \in [-1, -0.6]$$

在以后的搜索计算中,也把 w 的更新值限制在上述范围内,这样可以保证得到合理的最优权值向量。

3.3 适应度函数

和声搜索算法的适应度函数用于描述每组和声向量对应的权值向量 w 的优劣程度,采用推荐系统的平均绝对误差 MAE 作为权值向量的适应度函数。

MAE 通过比较真实评价值和预测值而得到,计算步骤如下:

(1) 通过计算相似度函数,获得用户 u 的最邻近的 k 个用户;

(2) 计算用户 u 对于项目 i 的预测评分^[14]:

$$P_u^i = \bar{r}_u + \frac{\sum_{n \in K_u} [\text{sim}_w(u, n) \times (r_n^i - \bar{r}_n)]}{\sum_{n \in K_u} \text{sim}_w(u, n)} \quad (2)$$

其中, \bar{r}_u 和 \bar{r}_n 是目标用户 u 和邻居用户 n 得到平均评分值, r_n^i 为邻居用户 n 对于项目 i 的评分值, K_u 是目标用户 u 的 k 个邻居集, $\text{sim}_w(u, n)$ 是目标用户 u 和邻居用户 n 的相似度。

(3) 最后计算平均绝对误差 MAE 值^[4]:

$$\text{fitness} = \text{MAE} = \frac{1}{U} \sum_{u \in U} \frac{\sum_{i \in I_u} |P_u^i - r_u^i|}{I_u} \quad (3)$$

其中, U 表示训练用户数量, I_u 表示训练项目数量, MAE 值越小,说明预测精度越高。

3.4 和声搜索操作

本文的和声搜索算法采用常见的操作^[13]: 和声库中选择、从和声库中选择后调音、在允许范围内随机产生。利用这三种经典的操作可以快速得到理想的优化结果。

(1) 选择和随机产生: 选择概率 HMCR 决定了是

否从和声库 HM 中选择变量来构成新的向量元素的概率,其变化范围是 0 到 1,一般比较靠近 1,本文设为 0.9。(1 - HMCR)是决定是否从上下限范围内随机产生新的变量的概率。选择和随机产生操作如下所示^[13]:

$$w'_{ij} = \begin{cases} w_{ij}^{\text{index}} = w_{ij}^{\text{rand}(1, \text{HMS})} & \text{概率在 } [0, \text{HMCR}] \text{ 之内} \\ \text{rand}(Lw_{j \in d}, Uw_{j \in d}) & \text{概率在 } [\text{HMCR}, 1] \text{ 之内} \end{cases} \quad (4)$$

其中, $Lw_{j \in d}$ 指向量组中第 j 个 w 值的下边界, $Uw_{j \in d}$ 指向量组中第 j 个 w 值的上边界。

(2) 调音:如果新的向量元素是从和声存储库中得到,那么该变量还要决定是否调音, PAR 是初始调音的概率,一般比较靠近 0,本文取值为 $[0.1, 0.4]$,便于其动态调整,概率 $(1 - \text{PAR})$ 是不需要调音的概率,调音是在原来的变量的基础上增加或减少一个随机带宽量 bw 。调音操作如下所示^[13]:

$$w'_{ij} = \begin{cases} w'_{ij} \pm \text{rand}(0, 1) \cdot bw & \text{概率在 } [0, \text{PAR}] \text{ 之内} \\ w'_{ij} & \text{概率在 } [\text{PAR}, 1] \text{ 之内} \end{cases} \quad (5)$$

上述三种方法产生的新的和声将被返回并用于更新和声库。

3.5 和声搜索参数设置的改进

从和声搜索算法描述可见,该算法的调音概率和随机带宽这两个参数设置都不能随着当前变量的变化情况而发生变化,而是采用某种固定的形式,造成和声搜索算法缺乏自适应性,不能很好地跳出局部最优,本文采用如下方法动态改变这两个参数:

调音的时候,可以动态地改变调音概率 PAR 和随机带宽量 bw 的值,以提高算法的收敛性。调音概率 PAR 的更新公式如下:

$$\text{PAR}_i = \text{PAR}_{\min} + i \times \frac{\text{PAR}_{\max} - \text{PAR}_{\min}}{T} \quad (6)$$

其中, i 是第 i 次迭代, T 是迭代的总数, PAR_{\max} 和 PAR_{\min} 分别是最大和最小的调音概率,这样算法迭代初期, PAR_i 比较小,运行若干次后,和声库里的向量有趋于相同的趋势,为了提高和声库向量的多样性,需要适当提高调音概率,上述公式能保证 PAR_i 趋于最大值 PAR_{\max} ,避免了算法早熟而收敛到局部极小。

随机带宽量 bw 的调节规则如下:

$$bw_i = \left| \sin\left(1000 \times \frac{i}{T}\right) \times e^{(-10 \times \frac{i}{T})} \right| \quad (7)$$

这种调节方式使得 bw 的值起初振荡,然后逐渐固

定下来,振荡部分逐渐受阻,最终 bw 趋于 0.000 1 附近,这种调节规则对于增加算法的收敛速度和精度很有作用,如图 1 所示:

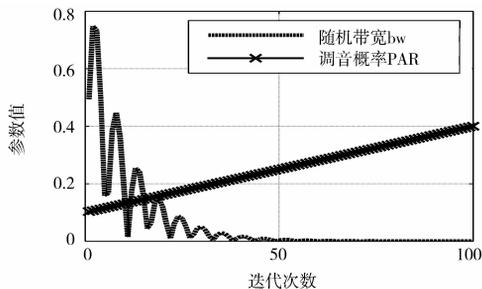


图 1 和声搜索算法的参数设置

这样调整之后,音符调节概率 PAR 与随机带宽 bw 都是变量。在算法开始运行时,改进后的和声搜索算法具有较小的 PAR 和较大的 bw ,这样该算法可在全局寻优;而当迭代多次后,算法具有较大的 PAR 和非常小的 bw ,以便在局部范围内寻找最优解。

3.6 更新和终止

每次迭代后将产生一组和声向量,如果新的和声向量在适应度上优于和声库中的最差的向量,那么就新的和声向量取代那个最差的向量。同时找出和声库的最佳适应度。由于是替换,因此每次迭代之后,和声库向量组的数量保持不变^[13]。

当和声库中的最佳适应度值低于某一阈值时,和声搜索过程终止,对于 MovieLens 数据库 ($m=1$ 和 $M=5$), 本文将阈值设为 0.78;或者当算法迭代次数达到 500 次时,计算过程也会终止。

4 实验设计

实验数据库是 MovieLens,采用两类实验来比较各种推荐系统,编程工具采用 Matlab7.1。

4.1 预测和推荐质量评价实验

采用两个推荐质量评价指标^[4]:平均绝对误差 MAE 和覆盖率 C 。

覆盖率是所有能够预测的项目占项目总数的比例,设为目标用户 u 提供的预测项集合为 $P_u = \{p_u^i | i = 1, 2, \dots, I_u\}$,其中 $p_u^i \neq 0$ 的个数为 N_u ,则推荐系统的覆盖率 C 定义为^[4]:

$$C = \frac{1}{U} \sum_{u \in U} \frac{N_u}{I_u} \quad (8)$$

文中提出的算法将和传统的协同过滤算法进行比较,比如余弦相似性(COS)^[4]、Pearson 相关系数(COR)^[5]和约束 Pearson 相关系数(CCOR)^[6]。

由于提出的和声协同过滤推荐算法是一种启发式搜索算法,具有一定的随机性,因此运行算法 20 次,以找出其中适应度最大和最小的两次运行结果,用于和传统的协同过滤算法进行比较。用本文的算法、COS、COR 和 CCOR 等几种协同过滤算法分别计算 MAE 和 C 这两个评价价值。

只用一部分数据来训练提出的算法以获得最优的相似度函数,一旦获得该函数,就用它来对测试数据进行预测和推荐,而不必再调用和声搜索算法来寻优。

对于 MovieLens 数据,随机选择 80% 的用户作为训练数据,剩下 20% 的用户作为测试数据。对于用户的最近邻居个数 k 的设定,取在 [10, 100] 之间。

4.2 性能实验

这类实验用于验证本文提出的算法是否比传统的相似度算法更加高效。从理论上讲,由于本文提出的相似度函数比传统的算法更加简单,因此,获得任意目标用户的 k 最邻近用户应该比传统算法更快,从而进行预测和产生推荐的速度也就更快。

5 结果分析

5.1 预测和推荐结果

将 MovieLens 数据库进行测试后,得到两个评价价值的结果。

(1)平均绝对误差 MAE: COS、COR 和 CCOR 以及本文的算法得到的 MAE 值中,本文算法对应的 MAE 值最小。MAE 的对比结果如图 2 所示:

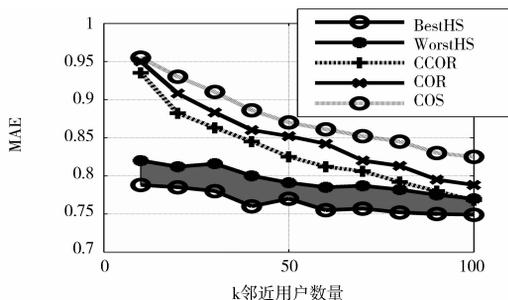


图 2 各种算法的平均绝对误差 MAE

(2)覆盖率 C:本文算法提高了 k 取值 [10, 100] 时的覆盖率 C,无论是适应度最大还是最小时,相比之

下,本文算法可以获得最佳的覆盖率。C 的对比结果如图 3 所示:

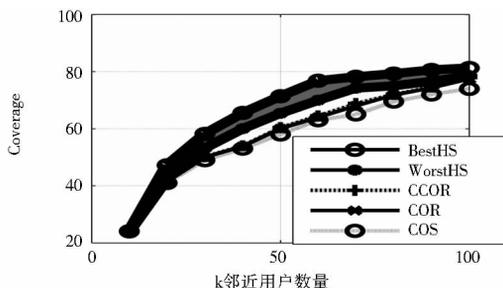


图 3 各种算法的覆盖率 C

综上所述,本文算法提高了预测精度和覆盖率,综合考虑 MAE 和覆盖率 C,当 k 取较大值时,可以同时获得最佳的 MAE 和覆盖率 C,这将提供更好的推荐效果。

5.2 性能结果

性能比较结果显示,本文算法比传统相似度算法所用的时间更少。各种算法的运行性能如表 1 所示:

表 1 各种算法的性能结果

相似度函数	COS	COR	CCOR	本文算法
时间消耗(秒)	25.23	26.42	22.54	13.65

为一个目标用户寻找 k 最邻近用户几乎占用了推荐系统的全部时间,因此简化用户之间的相似度函数的计算方法对于加速推荐过程来说就显得尤为重要。本文提出的采用自适应加权相似度函数简化了相似度计算公式,相应算法推荐速度比传统方法更快。

6 结语

协同过滤算法存在一些问题,如评分矩阵稀疏性、推荐准确性低、推荐效率性低。为解决这些问题,本文提出和声搜索算法来获得最优的协同过滤推荐系统的相似度函数,其关键技术在于:

(1)改进了相似度计算方法,该方法利用评分等级的差异进行加权统计,如公式(1)所示。避免了直接用评分矩阵来计算用户间相似度的方式,从而解决了协同过滤中评分矩阵稀疏性问题。

(2)提出了和声搜索算法的调音概率和随机带宽的量化计算方法,如公式(6)和公式(7)所示。可以避免和声搜索算法陷入局部极小,从而保证了该算法能找到最优的相似度权值向量。

对于标准数据库 MovieLens, 本文将提出的算法和经典的三种算法进行对比实验, 结果表明, 本文提出的算法提高了预测的精度和覆盖率, 为系统提供了更好的推荐效果。

由于本文只考虑了用户评分等级的差异, 接下来可以利用更多的客观信息来评价用户间的相似度, 进一步改进相似度计算方法, 如考虑评分等级的差异的均值和方差等因素。此外, 将该算法用于数字图书馆图书推荐系统中也是今后需要开展的工作。

参考文献:

- [1] Renda M E, Straccia U. A Personalized Collaborative Digital Library Environment: A Model and an Application[J]. *Information Processing and Management*, 2005, 41(1): 5 - 21.
- [2] Porcel C, Viedma H E. Dealing with Incomplete Information in a Fuzzy Linguistic Recommender System to Disseminate Information in University Digital Libraries[J]. *Knowledge - based Systems*, 2009, 23(1): 32 - 39.
- [3] 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. (Xu Hailing, Wu Xiao, Li Xiaodong, et al. Comparison Study of Internet Recommendation System[J]. *Journal of Software*, 2009, 20(2): 350 - 362.)
- [4] Sarwar B, Karypis G, Konstan J, et al. Item - based Collaborative Filtering Recommendation Algorithms[C]. In: *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, China. 2001: 285 - 295.
- [5] Resnick P, Iacovou N, Suchak M, et al. GroupLens: An Open Architecture for Collaborative Filtering of Netnews[C]. In: *Proceedings of ACM Conference on Computer Supported Cooperative Work*. New York: ACM Press, 1994: 175 - 186.
- [6] Adomavicius G, Tuzhilin A. Towards 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.
- [7] Kim K, Ahn H. A Recommender System Using GA K - means Clustering in an Online Shopping Market[J]. *Expert Systems with Applications*, 2008, 34(2): 1200 - 1209.
- [8] 张玲, 王磊, 王妹媛. 基于聚类免疫网络的协同过滤推荐算法[J]. *计算机工程与应用*, 2008, 44(27): 141 - 144. (Zhang Ling, Wang Lei, Wang Shuyuan. Clustering and Immune Mechanisms Based Collaborative Filtering Recommendation Algorithm[J]. *Computer Engineering and Applications*, 2008, 44(27): 141 - 144.)
- [9] 吴月萍, 王娜, 马良. 基于蚁群算法的协同过滤推荐系统的研究[J]. *计算机技术与发展*, 2011, 21(10): 73 - 76. (Wu Yueping, Wang Na, Ma Liang. Research of Collaboration Filtering Recommendation System Based on Ant Algorithm[J]. *Computer Technology and Development*, 2011, 21(10): 73 - 76.)
- [10] 高立群, 葛延峰, 孔芝, 等. 自适应和声粒子群搜索算法[J]. *控制与决策*, 2010, 25(7): 1101 - 1104. (Gao Liqun, Ge Yanfeng, Kong Zhi, et al. Adaptive Harmony PSO Search Algorithm[J]. *Control and Decision*, 2010, 25(7): 1101 - 1104.)
- [11] Yadav P, Kumar R, Panda S K, et al. An Improved Harmony Search Algorithm for Optimal Scheduling of the Diesel Generators in Oil Rig Platforms[J]. *Energy Conversion and Management*, 2011, 52(2): 893 - 902.
- [12] Askarzadeh A, Rezazadeh A. An Innovative Global Harmony Search Algorithm for Parameter Identification of a PEM Fuel Cell Model[J]. *IEEE Transactions on Industrial Electronics*, 2012, 59(9): 3473 - 3480.
- [13] Geem Z W, Kim J H, Loganathan G V. A New Heuristic Optimization Algorithm: Harmony Search[J]. *Simulation*, 2001, 76(2): 60 - 68.
- [14] Breese J, Hecherman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering[C]. In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*. 1998: 43 - 52.

(作者 E - mail: wanghuaqiu@163. com)