

基于用户模糊聚类的协同过滤推荐研究

李 华 张 宇 孙俊华

(重庆大学计算机学院 重庆 400044)

摘 要 传统的协同过滤算法没有考虑用户的自身信息对评分的影响,存在的数据稀疏性、扩展性差等弊端直接影响了推荐系统的推荐质量。对此提出了一种基于用户情景模糊聚类的协同过滤推荐算法。首先根据用户情景信息利用模糊聚类算法得到情景相似的用户群分类,然后在进行协同过滤前预先通过 Slope One 算法填充用户-项目评分矩阵,以有效改善数据稀疏性和实时性。实验结果表明,改进后的算法在推荐精度上有较大提高。

关键词 协同过滤,数据稀疏性,用户情景,模糊聚类,推荐

中图法分类号 TP391.7 文献标识码 A

Research on Collaborative Filtering Recommendation Based on User Fuzzy Clustering

LI Hua ZHANG Yu SUN Jun-hua

(College of Computer, Chongqing University, Chongqing 400044, China)

Abstract Traditional collaborative filtering algorithm does not consider the influence caused by the users' information, and the existing issues such as data sparsity, poor scalability and others directly affect the recommendation quality of recommendation systems. To address these issues, a collaborative filtering algorithm based on user context fuzzy clustering was proposed. First, users are clustered by fuzzy clustering algorithm according to user context, then the user-item rating matrix should be filled through slope one algorithm in advance before the traditional collaborative filtering. This effectively improves the sparsity of user rating data and the real-time performance. The experimental results indicate that the recommendation accuracy of the advanced approach is largely improved.

Keywords Collaborative filtering, Data sparsity, User context, Fuzzy clustering, Recommendation

1 引言

随着互联网的普及和计算机技术的迅速发展,个性化的推荐系统成为一个越来越受关注的研究领域。研究者提出多种推荐方法:基于内容的推荐、协同过滤推荐和混合推荐等,并结合先进的技术,如聚类、关联规则、贝叶斯网、神经网络和图模型等来实现这些方法^[1,2]。

协同过滤作为当前应用最成功的推荐技术,在很多领域中得到了实际的应用,但其仍存在许多问题需要解决。目前很多技术都是围绕协同过滤展开研究的,最典型的有稀疏问题(sparsity)和冷启动问题(cold-start)^[3,4],除此之外,还有新用户问题和算法扩展性等问题。协同过滤算法一般基于用户-项目评分矩阵,当该矩阵非常稀疏时,算法很难为目标用户找到相似用户,从而使推荐质量受到很大的限制。另一方面,协同过滤不关心资源和用户本身的信息,资源或用户语义上的层次信息和协同过滤推荐的用户相似度之间不存在关联性,而利用这些信息可以有效地提高协同过滤的推荐精度^[5]。推荐质量在很大程度上取决于近邻用户的选取的准确性,协同过滤的本质决定了要获得满意的推荐效果必须得到准确的用户信息。

本文引入用户情景,提出基于用户情景聚类的协同推荐。它首先利用模糊聚类算法将具有相似兴趣爱好的用户分配到相同的类中;然后在目标用的小的聚类中,根据 Slope One 算法来预测未评分数据,并将其值填充在用户-项目评分矩阵中;最后采用基于用户的协同过滤算法来计算用户间的相似性和形成最近邻居,并完成未评分数据的预测,产生 Top-N 推荐。聚类技术大大缩小了用户的最近邻居搜索范围,提高了推荐的实时性。同时,通过填充用户评分矩阵改进用户相似性的度量方法,有效地解决了数据极端稀疏性的问题。实验表明,该方法明显提高了推荐系统的推荐质量。

2 基于用户模糊聚类的协同过滤推荐

2.1 模糊聚类分析

模糊聚类分析的实质就是根据研究对象本身的属性来构造模糊矩阵,在此基础上根据一定的隶属度来确定其分类关系。模糊聚类分析算法具有分类简单且分类结果精度高的优点。

2.1.1 建立原始数据矩阵

论域 $U = \{x_1, x_2, \dots, x_n\}$ 为 n 个待分类的对象,每个对象有 m 个指标表示其性质, $U_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$, $i = 1, 2,$

收稿日期:2012-02-15 返修日期:2012-07-06 本文受国家自然科学基金项目(71102065)资助。

李 华(1962—),女,副教授,主要研究方向为计算机网络与通信,E-mail:lh@cqu.edu.cn;张 宇(1987—),女,硕士,主要研究方向为现代远程教育、数据挖掘;孙俊华(1987—),男,硕士,主要研究方向为 Web 数据挖掘、Web 智能检索。

..., n 。由此可得到原始数据矩阵

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

2.1.2 样本数据标准化

本文采用的是极差(极大极小)标准化,设标准化之后的样本集为 $X = (x_{ij}')_{n \times m}$, 则

$$x_{ij}' = \frac{x_{ij} - x_{j \min}}{x_{j \max} - x_{j \min}} \quad (1)$$

式中, $i = 1, 2, \dots, n, j = 1, 2, \dots, m$, 并且 $x_{j \min} = \min_{1 \leq i \leq n} \{x_{ij}\}$,

$$x_{j \max} = \max_{1 \leq i \leq n} \{x_{ij}\}。$$

实施数据标准化之后,每个指标值均在区间 $[0, 1]$ 中。

2.1.3 构造模糊相似矩阵

该操作又称标定,计算标定的方法很多,主要有欧式距离法、余弦夹角法、相关系数法、最大最小法、最小算术平均法、最小几何平均法等,具体内容可参考文献[6]。本文选用最大最小法来计算样本的相似性统计量 r_{ij} ,建立模糊相似矩阵 $R = (r_{ij})_{n \times m}$ 。计算公式为:

$$r_{ij} = \frac{\sum_{k=1}^m \min(x'_{ik}, x'_{jk})}{\sum_{k=1}^m \max(x'_{ik}, x'_{jk})} \quad (2)$$

式中, $i = 1, 2, \dots, n, j = 1, 2, \dots, m, x'$ 为上述标准化矩阵的元素。

2.1.4 建立模糊等价矩阵

从模糊相似矩阵 R 出发,采用平方法计算传递闭包 $t(R): R \rightarrow R^2 \rightarrow R^4 \rightarrow \cdots \rightarrow R^{2^k}$,经有限次运算后,一定有 $R^{2^k} = R^{2^{k+1}}$,于是 $t(R) = R^{2^k}$ 。 $t(R)$ 即为所求的模糊等价矩阵 R^* 。

2.1.5 模糊聚类

得到模糊等价矩阵 R^* 后,在适当水平 λ 上截取 R^* ,将模糊等价矩阵中大于值 λ 的数归为一类,就可以得到该水平下的聚类结果。

2.2 基于用户的协同推荐

协同过滤推荐根据其他用户的观点产生对目标用户的推荐列表,它基于这样一个假设^[7]:如果用户对一些项目的评分比较相似,则他们对其他项目的评分也比较相似。协同过滤推荐系统使用统计技术来搜索目标用户的若干最近邻居,然后根据最近邻居对项目的评分预测目标用户对项目的评分,产生对应的推荐列表。

2.2.1 用户相似度

为了计算用户对未评分项的预测评分,首先需要根据用户的历史评分记录计算用户间的评分相似度。度量用户之间相似性的方法主要包括如下3种^[8]:余弦相似性(cosine)(见式(3))、相关相似性(correlation)(见式(4))和修正的余弦相似性(adjusted cosine)(见式(5))。

$$\text{sim}(u, i) = \cos(\vec{u}, \vec{i}) = \frac{\vec{u} \times \vec{i}}{\|\vec{u}\| \times \|\vec{i}\|} \quad (3)$$

$$\text{sim}(u, i) = \frac{\sum_{j \in I_{ui}} (R_{u,j} - \bar{R}_u)(R_{i,j} - \bar{R}_i)}{\sqrt{\sum_{j \in I_{ui}} (R_{u,j} - \bar{R}_u)^2} \sqrt{\sum_{j \in I_{ui}} (R_{i,j} - \bar{R}_i)^2}} \quad (4)$$

$$\text{sim}(u, i) = \frac{\sum_{j \in I_{ui}} (R_{u,j} - \bar{R}_u)(R_{i,j} - \bar{R}_i)}{\sqrt{\sum_{j \in I_{ui}} (R_{u,j} - \bar{R}_u)^2} \sqrt{\sum_{j \in I_{ui}} (R_{i,j} - \bar{R}_i)^2}} \quad (5)$$

式中, $R_{u,j}$ 、 $R_{i,j}$ 分别表示用户 u 和用户 i 对项目 j 的评分, $\bar{R}_u$ 和 $\bar{R}_i$ 分别表示用户 u 和用户 i 对各自项目的平均评分。 I_{ui} 表示用户 u 和用户 i 共同评分的项目集合, I_u 、 I_i 分别表示用户 u 和用户 i 各自评分过的项目集合。

2.2.2 预测评分

在计算预测评分时,经常采用 K 近邻方法^[9],即选择与当前用户相似度最高的 K 个用户作为当前用户的邻居节点来进行计算。设 N_u 表示用户 u 的最近邻集合,利用式(6)预测目标用户 u 对未评分项 j 的评分值:

$$P_{u,j} = \bar{R}_u + \frac{\sum_{l \in N_u} \text{sim}(u, l) \times (R_{l,j} - \bar{R}_l)}{\sum_{l \in N_u} |\text{sim}(u, l)|} \quad (6)$$

式中, $\bar{R}_u$ 表示目标用户 u 对已评分项的平均评分值, $\text{sim}(u, l)$ 表示用户 u 与邻居 l 的相似度, $\bar{R}_{l,j}$ 表示邻居 l 对项目 j 的评分, $\bar{R}_l$ 表示邻居 l 对已评分项的平均评分值。

3 基于用户模糊聚类的协同过滤推荐的算法

基于用户模糊聚类的协同过滤算法的基本思想是:在进行传统的协同过滤前,先通过模糊聚类将情景相似的用户聚为一类来降低矩阵的维数,然后通过 Slope One 算法预测矩阵中的未评分项,构建一个不再稀疏的评分矩阵,对填充过的矩阵再进行协同过滤。比起传统的协同过滤算法,本文的协同过滤是在具有相似情景的用户群中进行的,有相似属性的人群更容易产生相似的爱好的,且协同过滤前先对稀疏矩阵进行了填充,这大大提高了近邻用户选取的准确性,提高了推荐的质量。同时,低维矩阵必定比高维矩阵的计算时间少,也提高了算法的可扩展性。另一方面,由于用户情景信息的相对稳定性,模糊聚类可以离线进行,不会造成在线推荐时的性能瓶颈。算法主要分为两步:对用户情景建模,利用模糊聚类算法得到情景相似的用户聚类;填充用户-项目评分矩阵,采用基于用户的协同过滤算法做出推荐。

3.1 用户情景相似度聚类

文献[10]提到80%的用户愿意向Web站点提供自己的性别、教育背景和兴趣等不太敏感信息,而且目前越来越多的网站都提供谷歌、雅虎、腾讯等账号登录服务,并可以提供更多更完善的用户档案信息。针对这些事实驱动,本文考虑利用已有的用户档案信息来对传统的协同过滤算法进行改进。

用户背景信息可以使用多种不同类型的属性来进行描述。一般包括连续的数值属性,如年龄;二元属性,如性别;分类(标称)属性,如学历、工作等。这些信息在系统初期能够实现辅助决策,部分解决了由于没有用户使用信息而导致的系统冷启动问题。首先根据用户背景信息进行用户情景建模,然后利用第2.1节介绍的模糊聚类算法将具有相似情景的用户分配到相同的类中。这里,设集合 $C = \{C_1, C_2, \dots, C_k\}$ 表示所有类型的集合,每个用户只能属于集合 C 中的一种类型。若 a_i, a_j 同属于类 C_k ,则 a_i, a_j 具有相似的用户情景,即他们的性别、年龄、职业等因素都非常接近,因而对同一事物也更

容易产生相似的喜好。

3.2 协同过滤推荐的改进算法

在上述第 3.1 节得到的聚类结果中,假设目标用户所在的类为 C_k ,对目标用户做出推荐,还需在类 C_k 中根据用户评分矩阵进行协同过滤。

在传统的相似性度量方法中,余弦相似性度量方法和修正的余弦相似性度量方法中对所有未评分项的评分均假设为 0,相关相似性度量方法中用户共同评分的项目集比较小。为了有效地解决用户评分数据极端稀疏情况下传统相似性度量方法存在的问题,本文在协同过滤前先通过 Slope One 算法对评分矩阵进行预测填充。

Slope One 是一种简单的基于项目的个性化推荐算法,在很多场合都能取得很好的效果。设系统中有 3 个用户 John、Jim、Tom,2 个项目 A 和 B。表 1 中, Tom 对项目 A 的评分就是需要预测的评分。根据 Slope One 的预测过程,首先需要计算平均偏离量 $dev_{(A,B)}$ 。

表 1 用户对项目评分表

	项目 A	项目 B
John	2	1
Jim	3.5	1.5
Tom	?	2

$$dev_{(A,B)} = \frac{(2-1) + (3.5-1.5)}{2} = 1.5$$

$$? = 2 + dev_{(A,B)} = 3.5$$

Slope One 的计算形式形如 $f(x) = x + b$ 。其中,变量 x 反映目标用户评分水平, b 是一个常量,表示目标项目的平均偏离量。假定项目 i 和 j 的用户评分向量为 $I = \{u_{i1}, u_{i2}, \dots, u_{in}\}$ 和 $J = \{u_{j1}, u_{j2}, \dots, u_{jn}\}$,然后根据最小二乘法进行线性拟合,可求得令目标函数 $\sum_k (u_k + b - u_{kj})^2$ 达到最小值的参数 $b = \sum_k (u_k - u_{kj}) / n$ 。

设经用户 u 和用户 i 评分的项目集合的并集 $I_u = I_u \cup I_i$,对于 I_u 中用户 u 未评分的项目 j 通过 Slope One 算法进行预测。首先利用式(7)计算项目 j 相对于项目 i 的平均偏离量 $dev_{(j,i)}$,然后利用式(8)预测用户 U 对项目 j 的评分值 $P(u_j)$ 。

$$dev_{(j,i)} = \sum_{u \in I(i) \cap I(j)} \frac{u_j - u_i}{num(I(i) \cap I(j))} \quad (7)$$

$$p(u_j) = \frac{1}{num(R(u) \cap D(j))} \sum_{i \in R(u) \cap D(j)} (u_i + dev_{(j,i)}) \quad (8)$$

式中, u_i 和 u_j 表示用户 U 对项目 i 和 j 给出的实际评分, $num(i \cap j)$ 表示同时对项目 i 和 j 给出评分的用户个数, $R(u)$ 代表目标用户 U 已经评过分的项集合, $D(j)$ 表示与目标项目 j 计算得到的平均偏离量项目的集合, $num(R(u) \cap D(j))$ 表示集合中元素的个数。

通过上述方法处理后,用户 u 和用户 i 对项目集合 I_u 中的所有项目均有评分,即对任意的项目 $j \in I_u$,用户 u 对项目 j 的评分为:

$$R(u_j) = \begin{cases} r(u_j), & \text{if user } u \text{ rated item } j \\ p(u_j), & \text{if user } u \text{ not rated item } j \end{cases}$$

在本文提出的方法中,相关相似性和修正的余弦相似性是一致的。对调整后的新矩阵,首先根据第 2.2.1 节的方法计算目标用户 u 和类 C_k 中其他用户间的相似度,取相似度最

高的用户作为最近邻居集 $N_u = \{N_{u1}, N_{u2}, \dots, N_{uk}\}$,且 u 与 N_{u1} 的相似性 $\text{sim}(u, N_{u1})$ 最高, u 与 N_{u2} 的相似性 $\text{sim}(u, N_{u2})$ 次之,依此类推。然后根据第 2.2.2 节对目标用户 u 产生 Top-N 推荐集。

3.3 推荐过程

传统的协同过滤算法使用余弦相似性、相关相似性等度量方法分析用户-项目评分矩阵中的向量,查找目标用户的最近邻,然后进行 Top-N 推荐。与此不同,本文提出的新方法首先采用模糊聚类算法将情景相似度高的用户聚为一类,据此可以得到用户-项目评分矩阵的子矩阵,该子矩阵中的用户是目标用户所在的聚类。在对该子矩阵搜索最近邻之前先对用户评分矩阵进行预处理,使其不再稀疏,然后进行协同过滤。新的算法流程如图 1 所示。

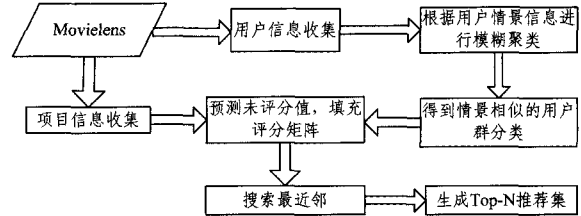


图 1 新算法流程图

输入:用户-项目评分矩阵 $R_{m \times n}$,用户情景信息表 X ,评分相似性阈值 r ,模糊聚类相似度阈值 λ 。

输出:目标用户对每个项目的预测评分 $P_{u,j}$, Top-N 推荐。

- (1)针对用户情景信息表 X ,利用式(1)对 X 中的数据进行标准化,得到样本矩阵 X' ;
- (2)利用式(2)计算样本矩阵 X' 中样本两两之间的相似度 r_{ij} ,得到模糊相似矩阵 R ;
- (3)采用平方法求 R 的模糊等价矩阵 R^* ;
- (4)调整阈值 λ ,得到最佳的聚类结果 $\{C_k\}, k = \{1, 2, \dots, p\}$;
- (5)从用户-项目评分矩阵 $R_{m \times n}$ 中获取子集 R_k, R_k 中的用户是聚类结果 $C_k, k = \{1, 2, \dots, p\}$ 中的用户,对 R_k 中的零值利用式(7)和式(8)预测填充得到矩阵 R_k' 。
- (6)在 R_k' 中利用式(4)求得目标用户 u 的最近邻居集合 $N_u = \{N_{u1}, N_{u2}, \dots, N_{uk}\}$;
- (7)根据最近邻居集 N_u 对项目的评分值,按照式(6)计算目标用户 u 对待推荐项的评分,产生 Top-N 推荐集。

3.4 算法复杂度分析

根据第 3.3 节算法的描述,设 m 为用户数, n 为项目数, r 为用户情景信息属性数, u 为目标用户,步骤(1)~(4)为模糊聚类过程,该过程要计算所有用户两两之间的相似度,因此复杂度为 $O(m \times m)$,步骤(5)为矩阵的预测填充,假设有 $p(p < m)$ 个用户对 n 个项目进行了评分,则每个用户各项目之间的偏离量计算需要 n^2 次,该步骤的时间复杂度为 $O(pn^2)$ 。步骤(6)和(7)的整体时间复杂度和传统的协同过滤算法相同,即为 $O(m \times n)$ 。因此,新算法的时间复杂度为 $O(pn^2)$,由于 p 是模糊聚类后产生的情景相似的同类用户的个数,是一个远小于 m 的常数,因此时间复杂度为 $O(n^2)$ 。一般地,有 $n < m$,所以,新算法降低了算法的时间复杂度,且步骤(1)~(4)是离线进行的,只需在系统空闲时执行一次。新算法通过聚类技术在情景相似的同类用户中搜索最近邻,在缩短了推荐时间的前提下,也保证了推荐的质量。

4 实验结果及其分析

4.1 数据集及度量标准

实验采用 MovieLens 站点 (<http://movielens.umn.edu>) 提供的数据集。该站点是一个基于 Web 的研究型推荐系统,用于接收用户对电影的评分并提供相应的电影推荐列表。该数据集中每个用户至少对 20 部电影进行了评分。整个实验数据集根据实验需要进一步地划分为训练集和测试集。其中训练集占总数的 80%,测试集占 20%。

采用平均绝对偏差 MAE 作为度量标准。平均绝对偏差 MAE 通过计算预测的用户评分与实际的用户评分之间的偏差来度量预测的准确性,MAE 越小,推荐质量越高。设预测的用户评分集合表示为 $\{p_1, p_2, \dots, p_N\}$,对应的实际用户评分集合为 $\{q_1, q_2, \dots, q_N\}$,则平均绝对偏差 MAE^[11] 定义为:

$$MAE = \frac{\sum_{i=1}^N |p_i - q_i|}{N} \quad (9)$$

4.2 实验结果

实验中首先根据用户情景信息对所有用户进行模糊聚类,选取最佳聚类阈值 $\lambda=0.7$,并以相关相似性作为寻找最近邻的相似度计算方法。将本文算法与传统的协同过滤推荐算法以及文献[12]提出的一种改进的协同过滤推荐算法进行对比,在计算 MAE 时,邻居个数从 5 个增至 30 个,间隔为 5。实验结果对比如图 2 所示。

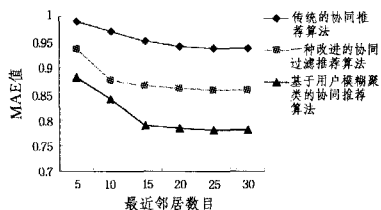


图2 实验结果图

从图2可以看出,随着目标用户的最近邻居数目的增加,上述3种算法的MAE值都呈现逐渐下降的趋势。但是在各种实验条件下,本文提出的改进算法均具有最小的MAE值。由此可知,本文提出的基于用户模糊聚类的协同过滤推荐算法可以有效地缓解评分矩阵稀疏的问题,提高推荐系统的推荐质量。

4.3 实验结果分析

传统的协同过滤推荐算法是在整个用户空间上根据所有用户对项目的评分来计算用户间的相似性,不仅计算量大,而且将所有用户对项目评分的重要性视为同等的,忽略了不同特征用户之间评分的差异性,这无疑会降低系统推荐的准确性。一种改进的协同过滤推荐算法只考虑了用户评分对项目相似性的影响,而没有充分利用已有的用户数据和领域知识,

特别是在评价数据稀疏的情况下,造成项目相似性计算不准确的问题,严重影响了推荐的精确度。而本文充分考虑了用户自身信息对评分的影响,协同过滤是在具有相似情景的同类用户中进行的,同类用户更容易产生相似的兴趣爱好,同时,在协同过滤前对稀疏评分矩阵进行了填充,这大大提高了近邻用户选取的准确性。实验结果表明,本文提出的算法有效地提高了推荐系统的推荐质量。

结束语 随着推荐系统规模越来越大,用户数目和项目数目急剧增加,推荐系统的实时性要求越来越难以满足。本文提出的算法引入用户情景因素,通过模糊聚类技术对稀疏用户-项目评分矩阵进行降维,然后对降维后的评分矩阵进行填充,最后利用协同过滤在线进行推荐。把聚类分析用于协同过滤推荐中,既降低了用户空间的维度,使得搜寻最近邻居用户的范围缩小,又能够提高协同过滤算法的可扩展性和效率。实验结果表明,基于用户模糊聚类的协同过滤算法有效地解决了数据稀疏性问题,提高了推荐的准确度。

参考文献

- [1] 李聪,梁昌勇,马丽. 基于领域最近邻的协同过滤推荐算法[J]. 计算机研究与发展,2008,45(9):1532-1538
- [2] 张光卫,康建初,李鹤松,等. 面向场景的协同过滤推荐算法[J]. 系统仿真学报,2006,18(z2):595-601
- [3] 郑先荣,曹先彬. 线性逐步遗忘协同过滤算法的研究[J]. 计算机工程,2007,33(6):72-74
- [4] Balabanovic M, Shoham Y. Fab: Content-based Collaborative Recommendation[J]. Communication of the Association for Computing Machinery,1997,40(3):66-72
- [5] 秦围,杜小勇. 基于用户层次信息的协同推荐算法[J]. 计算机科学,2004,31(10):138-140
- [6] 杨虎,刘琼荪,钟波. 数理统计[M]. 北京:高等教育出版社,2004:190-199
- [7] Breese J, Hecherman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[C]// Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98). 1998:43-52
- [8] Sarwar B, Karypis G, Konstan J, et al. Item-Based collaborative filtering recommendation algorithms[C]// Proceedings of the 10th International World Wide Web Conference. 2001:285-295
- [9] 罗辛,欧阳元新,熊璋,等. 通过相似度支持度优化基于K近邻的协同过滤算法[J]. 计算机学报,2010,33(8):1437-1445
- [10] 曾春,邢春晓,周立柱. 个性化服务技术综述[J]. 软件学报,2002,13(10):1952-1961
- [11] 张丙奇. 基于领域知识的个性化推荐算法研究[J]. 计算机工程,2005,31(21):7-9
- [12] 刘旭东,陈德人,王惠敏. 一种改进的协同过滤推荐算法[J]. 武汉理工大学学报:信息与管理工程版,2010,32(4):550-553
- [3] 杨金福,宋敏,李明爱. 一种新的基于距离加权的模板约简K近邻算法[J]. 电子与信息学报,2011,33(10):2378-2383
- [4] 童先群,周忠盾. 基于属性值信息熵的KNN改进算法[J]. 计算机工程与应用,2010,46(3):115-117
- [5] 牟琦,陈艺坤,毕孝儒. 一种基于快速增量SVM的入侵检测方法[J]. 计算机工程,2012,38(12):92-94
- [6] Fayed H A, Atiya A F. A novel template reduction approach for the k-nearest neighbor method[J]. IEEE Transactions on Neural Networks,2009,20(5):890-896
- [7] 夏书银,王越,张权. 核空间结合样本中心角度的支持向量机增量算法[J]. 计算机应用与软件,2012,4:121-124
- [8] KDD99Cupdataset[EB/OL]. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>,2011-01-14
- [9] 肖敏,柴蓉,杨富平,等. 基于可拓集的入侵检测模型[J]. 重庆邮电大学学报:自然科学版,2010,22(3):345-349

(上接第78页)