计算机研究与发展
Journal of Computer Research and Development
ISSN 1000-1239/CN 11-1777/TP
48(Suppl.): 205-212, 2011

基于协同过滤与划分聚类的改进推荐算法

吴泓辰[1] 王新军[1] 成 勇[2] 彭朝晖[1]
[1](山东大学计算机科学与技术学院 济南 250101)
[2](人力资源和社会保障部信息中心 北京 100716)
(hc_wu@mail.sdu.edu.cn)

Advanced Recommendation Based on Collaborative Filtering and Partition Clustering

Wu Hongchen[1], Wang Xinjun[1], Cheng Yong[2], and Peng Zhaohui[1]
[1](*School of Computer Science and Technology, Shandong University, Jinan* 250101)
[2](*Information Center of Ministry of Human Resources and Social Security of the People's Republic of China, Beijing* 100716)

Abstract In this paper, an advanced recommendation algorithm used for personalized service has been put forward which is based on collaborative filtering and partition clustering. Firstly, a computing matrix used for modified collaborative clustering is built up which can be referred to give recommendation. Secondly, the range of setting value in matrix has been expanded to integrate all the users' evaluation. Finally, evaluating number and updating coefficient are created to make the update available and thus gives recommendations to users which can satisfy them. Furthermore, a modified partition clustering is proposed which in next step enhance the accuracy and real-time of recommendation algorithm. In the end, we experimentally proved the ultimate recommendation algorithm based on modified partition clustering is the utmost solution for personalized service, which can provide the most satisfactory information to users.

Key words personalized service; advanced recommendation; collaborative filtering; partition clustering

摘 要 针对个性化服务技术提出一种改进推荐算法,该方法基于协同过滤技术和划分聚类技术. 首先建立了协同过滤推荐算法的计算矩阵,使算法能够参照矩阵来推荐信息,其次完善了矩阵的赋值范围,使推荐算法能综合所有用户的评价,最后添加了评价数值和更新系数,把算法的动态更新变为可能,从而推荐给用户最满意的信息. 在此基础上还提出基于划分聚类的改进推荐算法,进一步提高了算法的准确性和实时性,并且用实验证明了基于划分聚类的最终推荐算法是最优的个性化服务推荐算法,能够提供给用户最满意的推荐信息.

关键词 个性化服务技术;改进推荐算法;协同过滤;划分聚类

中图法分类号 TP311

收稿日期:2011-07-15

基金项目:国家科技支撑计划基金项目(2009BAH44B01);国家自然科学基金项目(61003051);山东省科技攻关计划基金项目(2010GGX10114,2010GGX10108);山东大学自主创新基金项目(2010TS057)

个性化服务(personalized service)是 Web 信息处理技术的主要发展趋势之一[1-5]. 随着互联网技术的飞速发展和日渐完善,网络中的信息量也在爆炸式地增多,虽然搜索引擎已经成为广大用户在海量数据中搜索信息的最有效办法,但它的大众性并不能满足每个用户的搜索习惯、搜索喜好等方面的要求,个性化服务技术因此而诞生,它通过收集和分析用户的信息,在用户手动搜索之前将物品推荐给用户,从而具备了主动推荐的能力.

个性化服务技术的核心在于它的推荐能力,如何使用恰当的推荐算法提升推荐信息的准确率,提高与用户感兴趣信息的吻合度,成为个性化服务技术的瓶颈. 为了更好地实现个性化服务技术,人们不断寻求更新的推荐算法,使其日趋完善. 目前主要的推荐算法有 3 类:第 1 类是基于序列的个性化推荐算法[6-7],主要通过分析用户的各项属性和使用习惯来制定合适的序列,这个序列通过"如果然后"的顺序分析出用户接下来需要什么从而完成推荐,这项技术很明显不是针对用户而是方便管理员使用的,程序开发出来以后管理员只需在开始时嵌入恰当的序列即可投入使用,很明显不能满足不同用户的需要,推荐准确率低;第 2 类是基于用户和资源关联的个性化推荐算法[8-9],主要通过分析互联网中用户与资源之间的关联来制定匹配对,过滤出用户和资源之间的相似性,虽然这类推荐算法在一定程度上比基于序列的个性化推荐算法更能满足用户的需求,但是在程序开发的初期缺乏用户和资源之间的关联数据,加大了管理员的工作量,而且网上的资源冗余量大,噪音很多,难以去除无人问津的资源,动态更新更难做到,基于以上两种个性化推荐算法,如何找到一种方法既能满足用户需求,又能减少管理员的工作量呢;第 3 类基于协作过滤的推荐算法[10-12]就是在这样的问题下产生的,它充分利用相似用户之间的共同特点,为用户提供没有访问但其他用户已经访问过的资源信息,推荐准确率高;管理员只需将相似用户分于同一组即可,无需大量的工作,因此基于协同过滤的推荐算法已逐步成为个性化服务技术的主流[13-16]. 其优点是容易发现用户感兴趣的信息,从而达到提升网站访问量的效果. 但协同过滤推荐算法面临的挑战依然存在,其主要难点有:1)可用数据的稀疏性:互联网拥有数以百万计的用户和待推荐的物品,但用户和这些物品之间的关联,例如喜欢、忌讳等信息却少之又少,这些大都需要从用户处收集,然而用户对于自己的隐私过于敏感,他们可能会提供性别、年龄等表层信息,因此能够用于表示用户兴趣的信息实际上是非常稀疏甚至有限的;2)数据的完整性:在大量数据当中,有的数据是真实可信的,方便提取和用户参考. 也有数据是错误的、缺失的、不完整的,如何在这些数据当中消除冲突、去伪存真成为数据处理技术的一大难点;3)互联网数据的动态性:在互联网方面,不断有新的数据加入和旧数据的删除;在用户方面,用户的兴趣和关注点也在不断地改变,用户在使用的过程中还在不断增加新的训练数据,这就要求推荐算法能够快速、准确地进行更新,实时性尤为重要.

为了更好地解决 Web 信息处理中所面临的上述问题,本文从分析个性化服务推荐技术的相关原理出发,建立改进协同过滤算法的计算矩阵,并给出评价数值和更新系数,使改进推荐算法能综合所有用户的评价,为用户进行满意有效的推荐. 在此基础上还提出基于划分聚类的改进推荐算法,进一步提高了推荐技术的准确性和实时性,并且用实验证明了基于划分聚类的最终推荐算法是最优的个性化服务推荐算法,能够提供给用户最满意的推荐信息.

1 相关工作

1.1 协同过滤推荐算法

在基于 Web 的个性化服务中,协同过滤推荐算法的基本原理是根据相似用户的兴趣来推荐当前用户没有看过但是很有可能会感兴趣的信息,所基于的假设是如果两个用户兴趣类似,那么很有可能当前用户会喜欢另一个用户所喜欢的内容,这不仅无需考虑资源的表示形式、不受推荐物品的具体内容限制,而且具有相当高的准确性. 下面根据协同过滤推荐算法的基本原理[10,17]说明协同过滤推荐算法.

步骤 1. 推荐系统中存在两个集合,一个是用户集合 $User$ 总数是 x,编号从 $1\sim x$;另一个是待推荐的物品集合 $Product$ 总数是 y,编号从 $1\sim y$,分别记作 $U(x)$和 $P(y)$. 现将这两个集合合并起来,排成一个 $x\times y$ 阶矩阵 $\boldsymbol{B}$,行向量是用户集合 U,列向量是物品集合 P,矩阵中共计元素数目是 $x\times y$.

步骤 2. 矩阵中元素只能取 0 和 1 这两个值来区分用户是否访问过该物品,假如矩阵中第 2 行 3 列的元素 $B[2][3]$赋值为 1,就代表用户 2 对物品 3 访问过,否则赋值为 0. 初步统计过后得到一个 0-1 矩阵 $\boldsymbol{B}$ 如式(1)所示:

$$R_{x\times y}=\begin{pmatrix}00101 & \cdots & 001110\\ 01100 & \cdots & 101001\\ \vdots & & \vdots\\ 00101 & \cdots & 101000\end{pmatrix}. \quad (1)$$

接下来就可以参照矩阵 B 完成物品推荐,将矩阵中的每一行看作一个用户是否对相应物品进行了访问,记为一个元组. 通过元组之间的对比,将该用户没有访问过但其他用户已经访问过的物品推荐给这个用户. 算法是遍历矩阵第 1 列到最后一列,每一列从上到下查找每一个元素,只要找到一个 1,就将所该列对应的物品推荐给该列中行值是 0 的用户,以下给出代码实现:

算法 1. Recommendation Algorithm according to 0-1 Matrix B.

① Input:
② $U(x)$ /*$User$ Group*/
③ $P(y)$ /* $Product$ Group*/
④ Output:
⑤ Offer proper products to users
⑥ Begin
⑦ Initial:
⑧ $i=1$ /*sentry of row vector*/
⑨ $j=1$ /* sentry of column vector*/
⑩ for each row vector do
⑪ repeat
⑫ if find j such that $B[i][j]=1$ then
⑬ for all $User(*)$ such that $B[i][*]=0$
⑭ offer $P(j)$ to $User(*)$
⑮ else $i+1$; /*change next row vector*/
⑯ until the last row vector is finished
⑰ End

1.2 现有算法的不足

0-1 矩阵 B 能够清晰地表现出用户是否访问过相应的物品,但是局限性也十分明显,主要表现在两个方面:第一,数值 0 和 1 只能看出用户是否访问过该物品,而不能够看出用户对该物品的喜好程度,这对于以后其他用户的访问没有任何参考价值;第二,算法存在无法动态更新的问题,一旦有新用户和推荐物品加入,现有的推荐方法无法做到与实际情况相吻合,从而导致推荐的物品在很大程度上偏离了用户的需求.

为此我们提出了协同过滤的改进推荐算法(advanced recommendation algorithm)和最终推荐算法(ultimate recommendation algorithm),使算法更偏向用户并添加了动态更新功能,消除了传统推荐算法的弊端.

2 基于协同过滤的改进算法描述

在 0-1 矩阵 B 元素的基础上加以改进,让每个矩阵元素取值不光取 0 和 1,还可以取 1 以上的值,值越高说明用户对该物品的评价越好,越值得被推荐,通过设定用户对物品的评价数值,不仅可以获得用户对物品的喜好程度,还可让用户自行设定推荐阈值,当评价数值低于自己设定的推荐阈值时不必推荐给自己,评价数值的计算可以由管理员分析后确定. 以下给出改进以后的算法描述.

设推荐系统中存在两个集合:一个是用户集合 $User$ 总数是 x,编号从 $1\sim x$;另一个是待推荐的物品集合 $Product$ 总数是 y,编号从 $1\sim y$,分别记作 $U(x)$ 和 $P(y)$. 现将这两个集合合并起来,排成一个 $x\times y$ 阶矩阵 B,行向量是用户集合 U,列向量是物品集合 P,矩阵中共计元素数目是 $x\times y$. 设定一个一维递增数组 $Num[0,1,2,\cdots,I_{\max}]$,数组元素取值从 0 到 $I_{\max}$ 共计 $I_{\max}+1$ 个元素,将 $x\times y$ 阶矩阵中元素按照用户对物品的喜好程度,用数组里的数进行赋值,值越大表示用户的评价越高,越值得推荐. 如果允许用户设定阈值 $Lim(i)$,$i\in(1,x)$,前提必须对每个物品添加评价数值,设为 $Value(j)$,$j\in(1,j)$. 以下给出代码的改进算法:

算法 2. Advanced Recommendation Algorithm.

① Input:
② $U(x)$ /*$User$ Group*/
③ $P(y)$ /*$Product$ Group*/
④ Output:
⑤ Offer proper products to users
⑥ Begin
⑦ Initial:
⑧ $i=1$ /*sentry of row vector*/
⑨ $j=1$ /*sentry of column vector*/
⑩ k /*inspector of sentry*/
⑪ for each row vector do
⑫ repeat
⑬ if find j such that $B[i][j]\geqslant 1$ then
⑭ for all $User(*)$ such that $B[i][*]=0$

⑮ if $Value(j) \geqslant Lim(i)$
⑯ /* satisfy user request */
⑰ offer $P(j)$ to $User(i)$
⑱ else continue /* not satisfy user request */
⑲ else $i+1$; /* change next row vector */
⑳ until the last row vector is finished
㉑ End

虽然评价数值的添入更有利于用户的自行管理,但是互联网中的数据是在不断更新的,用户在接到推荐算法提供的物品以后也可以对该物品给出自己的评价数值,而这个数值就有可能和其他用户给出的数值有差异甚至是截然相反. 为此,就必须给出评价数值 $Value(j)$ 的更新算法,并在数组 Num 中添加负值能够让用户表示对该物品不满意度,这样推荐算法就可以综合所有用户的评价,为其他用户提供满意的物品. 以下给出协同过滤最终推荐算法的描述.

设推荐系统中存在两个集合:一个是用户集合 $User$ 总数是 x,编号从 $1\sim x$;另一个是待推荐的物品集合 $Product$ 总数是 y,编号从 $1\sim y$,分别记作 $U(x)$ 和 $P(y)$. 现将这两个集合合并起来,排成一个 $x\times y$ 阶矩阵 $\boldsymbol{B}$,行向量是用户集合 U,列向量是物品集合 P,矩阵中共计元素数目是 $x\times y$. 设定一个一维递增数组 $Num[I_{\min}\cdots -1,0,1,2,3,\cdots,I_{\max}]$,其中 0 表示用户没访问该物品,负值可以表示对该物品没兴趣或不喜欢,数值越大表示用户评价越高,$I_{\min}$ 和 $I_{\max}$ 可以由推荐系统管理员设定. 评价数值的更新公式给出:

$$Value(j)=\alpha\times B[i][j]+(1-\alpha)\times Value(j). \quad (2)$$

式(2)中 $B[i][j]$ 表示将推荐物品提供给用户后用户给出的评价,α 是更新系数,取值范围是 $0\sim 1$,α 的计算公式如下给出:

$$\alpha=\frac{1}{1+userCounter}. \quad (3)$$

式(3)中的 $userCounter$ 表示已经对该物品进行评价的用户数目,α 的大小直接决定新的推荐数值更侧重于原有推荐数值还是该用户的评价,例如前面没有用户对该物品评价,那么 $userCounter=0$,$\alpha=1$,因此 $Value(j)=B[i][j]$;当 α 取值是 0.5 时,说明原 $Value(j)$ 和 $B[i][j]$ 对新 $Value(P)$ 的影响权重相同. 现给出最终代码如下所示:

算法 3. Ultimate Recommendation Algorithm.

① Input:
② $U(x)$ /*User Group*/
③ $P(y)$ /*Product Group*/
④ Output:
⑤ Offer proper products to users
⑥ Begin
⑦ Initial:
⑧ $i=1$ /*sentry of row vector*/
⑨ $j=1$ /*sentry of column vector*/
⑩ k /*inspector of sentry*/
⑪ $userCounter=0$ /*count the number of users*/
⑫ $\alpha=\frac{1}{1+userCounter}$
⑬ for each row vector do
⑭ repeat
⑮ if find j such that $B[i][j]\geqslant 1$ then
⑯ for all $User(*)$ such that $B[i][*]=0$
⑰ if $Value(j)\geqslant \mathrm{Lim}(i)$
⑱ /*satisfy user request*/
⑲ offer $P(j)$ to $User(i)$
⑳ $Value(j)=\partial\times Value(j)+(1-\partial)\times B[k][j]$
㉑ /*立刻更新评价数值*/
㉒ else continue /*not satisfy user request*/
㉓ else $i+1$; /*change next row vector*/
㉔ until the last row vector is finished
㉕ End

上文中的协同过滤推荐算法解决了相似用户之间的物品推荐,充分考虑到用户对于推荐物品的评价范围并做到了时刻更新,具有良好的实时性和准确性. 但是这都要在相似用户之间才能实现,比如把喜欢厨艺的用户的信息推荐给喜欢拳击的用户肯定不合适,准确率必然下降. 因此,将哪些相似的用户放在同一推荐系统中成为上述推荐算法的关键,而本文提出的划分聚类算法有效地解决了这一问题.

3 基于划分聚类的改进推荐算法

聚类是在没有人工标注的基础下,将具有相似属性的数据聚集在一起的无监督学习方法. 它使具备一定相似性的数据实例组织成一些相似组,处于同组内的数据彼此相似,处于不同组的数据彼此不

同. 协同过滤推荐算法就是要在相似度高的用户分组基础之上才能完成高效的物品推荐,聚类算法的好坏直接决定下一步协同过滤的效果和性能. 相似性计算直接制约聚类的效果,进一步影响整个推荐算法. 传统的聚类算法中,最常用的是 k 核心聚类,其简洁和高效是它被广泛使用的原因,下面给出其原有的算法原理:假定有一些用户的集合 $User$,用户总数是 m 记作 $User(U_1,U_2,U_3,\cdots,U_m)$,每个用户 U_x 各有 n 项属性,可以用一个向量来表示,记作 $U_x(C_{x1},C_{x2},C_{x3},\cdots,C_{xn})$. 聚类的原理就是在集合 $User$ 的基础上,按照用户的属性对比完成相似用户的分组. k 核心聚类核心思想是将给定的集合 $User(U_1,U_2,U_3,\cdots,U_m)$ 划分成 k 个相似组,k 的数目可以由管理员设定,每个组内设定一个聚类中心,也就是本组聚类中的均值,凡是与此均值差距较小或类似的用户可以分到本组之中,k 核心聚类的算法如下:

算法 4. Algorithm for k-centers clustering(k, $User$).

① Select k user points as the cluster center
② repeat
③ for each user data point $u\in User$ do
④ compute the distance from u to each center
⑤ merge u into the closest center
⑥ endfor
⑦ re-compute the center by using the current cluster
⑧ until the finishing criterion is met

在上述算法开始时,随机选取 k 个用户作为 k 个组的聚类核心,然后计算每个用户属性与核心属性之间的距离,接着分配到与其距离最小的核心所在的聚类中. 当所有用户都分配完毕以后,整个算法将被重新执行,直到所有用户每次都会被分配到一个固定的组中或没有一个核心发生变化. 分到该组的用户的属性与均值的属性差距越小越能表示相似度越高,越适合放在同一聚类中. 但是 k 核心聚类算法存在很多局限,k 的值是按照管理员主观得出而不是分析当前用户分布的实际情况;任意选取并反复寻找核心用户会使算法的复杂度过高;孤立用户会使得算法效果下降. 本文提出的划分聚类算法有效地解决了上述问题,其描述如下:假定有一些用户的集合 $User$,用户总数是 m 记作 $User(U_1,U_2,U_3,\cdots,U_m)$,每个用户 U_x 各有 n 项属性,可以用一个向量来表示,记作 $U_x(C_{x1},C_{x2},C_{x3},\cdots,C_{xn})$.

步骤 1. 预处理孤立用户. 孤立用户是指各项属性与最近用户差距远大于其他用户之间差距的用户,若将孤立用户与其他用户分在同一组中肯定会影响聚类的效果,为此必须将孤立用户找出并单独分在一组. 孤立用户的处理过程如下:

设用户总数是 m,因此每个用户与其他用户之间的路径共有 $m-1$ 个,m 个用户的路径总数:

$$L=\frac{m\times(m-1)}{2}, \tag{4}$$

则所有用户之间的距离之和为

$$D=\frac{1}{2}\times\sum_{i=1}^{m}\sum_{j\neq i}gap(C_i,C_j), \tag{5}$$

式(5)中的 m 代表用户总数,i 代表当前用户号,j 代表其他用户号,$gap(C_i,C_j)$ 代表用户 C_i 和 C_j 之间的距离,计算公式为

$$gap(C_i,C_j)=\sqrt{(C_{i1}-C_{j1})^2+(C_{i2}-C_{j2})^2+\cdots+(C_{in}-C_{jn})^2}, \tag{6}$$

式(6)中 $\{C_{i1},C_{i2},\cdots,C_{in}\}$ 是用户 C_i 的 n 个属性,$\{C_{j1},C_{j2},\cdots,C_{jn}\}$ 是用户 C_j 的 n 个属性. 设定参数极限均值(extreme mean value, EMV),依据式(4)和式(5)得出:

$$EMV=\frac{L}{D}. \tag{7}$$

对于任意一个用户 C_x,$x\in(1,m)$,如果该用户与其他所有用户的距离 $Line(C_x,C_y)$,($y\in(1,m)$,$x\neq y$),都有 $Line(C_x,C_y)\geqslant EMV$,则将用户 C_x 判定为孤立用户,单独划作一个聚类组.

步骤 2. 计算聚类的总数 k. 设用户的总数是 m,所基于的假设是,要么所有的用户分在一个聚类中,则 $k=1$;要么每个用户处于自己独立的聚类中,则 $k=m$;因此 $k\in(1,m)$.

定义引入参数最小距离方差(minimum distance variance, MDV):

$$\mathrm{MDV}=\sum_{j=1}^{k}\sum_{x\in C_j}dis(m,U), \tag{8}$$

式(8)中 k 代表的是聚类总数,j 表示聚类组号,x 表示分到聚类 C_j 中的用户,其中 $dis(m,c)$ 的计算表达式为

$$dis(m,c)=\sqrt{(m_1-U_1)^2+(m_2-U_2)^2+\cdots+(m_n-U_n)^2}, \tag{9}$$

式(9)中 $m\{m_1,m_2,\cdots,m_n\}$ 是聚类 C_j 核心用户的属

性向量,$U\{U_1,U_2,\cdots,U_n\}$代表分配到 C_j 用户的属性向量. 如果 k 的取值从 1 开始到 m,总共 m 组,组与组之间相互独立,各自计算组内的 MDV 数值,全部计算完毕以后,将 m 组的 MDV 相互比较,将最小数值作为最终 MDV,记录此时 k 的值 k_x,$k_x\in(1,m)$,并将其对应聚类分组作为最优解.

步骤 3. 验证正确性. 添加参数均值 m_l,$l\in(1,k_x)$,在已知聚类数为 k_x 的基础上,m_l 表示每个聚类组中所有用户的均值点,m_l 由式(10)计算:

$$m_l=\frac{1}{|C_l|}\sum_{x_l\in C_l}x_l, \tag{10}$$

式(10)中 $l\in(1,2,\cdots,k_x)$,$|C_l|$代表用户的总数,现在将每个聚类组的均值和聚类中的用户放在一起,组成$(m+k_x)$个用户,重复执行步骤 1 和步骤 2 可以得出同样的聚类分组. 以下用实例说明算法过程:

假定有一些用户的集合 $User$,用户总数是 12 记作 $User(U_1,U_2,U_3,\cdots,U_{12})$,将所有用户按照各自的属性排列在数轴当中,如图 1 所示:

图 1 随机选取的 12 个用户排列

k 的值从 1～12 循环计算 MDV,通过算法寻找每个取值时各自最理想的用户聚类方案,现设循环至 $k=2$,每个用户被分配给距离它最近的核心用户形成两个聚类,用 C_1 和 C_2 表示,m_1 和 m_2 分别是 C_1 和 C_2 的均值. 第 1 次循环可能得到的两个聚类如图 2 所示,两个随机确定的均值用“+”表示:

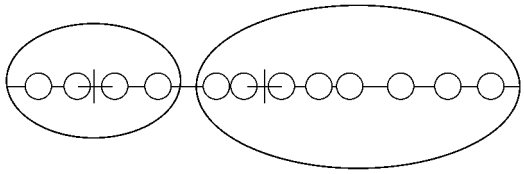

图 2 第 1 次循环聚类

该聚类结果未必是 $k=2$ 的最优聚类,接下来根据当前聚类中的用户属性值重新寻找均值并计算 MDV,进入下一循环,最终找到 $k=2$ 时的最小 MDV,对应聚类方案如图 3 所示:

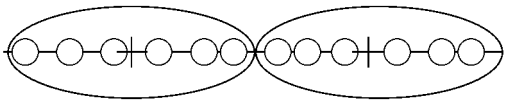

图 3 最终聚类

假设 $k=2$ 时所得到 MDV 是 k 这 12 个循环取值的最小 MDV,则 $k=2$ 时的 MDV 是最终 MDV,对应的聚类分组方案是最优解.

4 实验和评价

由于推荐算法是要建立在同组用户基础之上才能完成,而分组的任务是由聚类完成的,因此要衡量整个算法的好坏必须将两步结合起来,先用聚类找到相似用户,再使用推荐算法在相似用户分组之间完成推荐任务. 推荐算法在 1 节中介绍了 3 种,分别是推荐算法(recommendation algorithm, RA)、改进推荐算法(advanced recommendation algorithm, ARA)和最终推荐算法(ultimate recommendation algorithm,URA);聚类算法在 2 节中介绍了两种,分别是传统 k 核心聚类(k-means clustering,KMC)和改进的划分聚类(modified partition clustering, MPC). 在本节中,我们将通过实验验证推荐算法和聚类算法的性能好坏. 首先,为了比较推荐算法的好坏,给出了基于划分聚类的推荐算法(recommendation algorithm based on modified partition clustering, MPCRA)、基于划分聚类的改进推荐算法(advanced recommendation algorithm based on modified partition clustering, MPCARA)和基于划分聚类的最终推荐算法(ultimate recommendation algorithm based on modified partition clustering, MPCURA)的性能比较. 然后,为了比较聚类算法的好坏,给出了基于传统 k 核心聚类的最终推荐算法(ultimate recommendation algorithm based on k-means clustering, KMCURA)和基于划分聚类的最终推荐算法(ultimate recommendation algorithm based on modified partition clustering, MPCURA)的性能比较.

4.1 数据集

选用本课题组开发的公共就业服务信息网(http://211.87.239.93:1111/)的数据作为数据集,实验环境:Windows XP,Myeclipse 6.5,奔腾(R)4 处理器,主频 2.80 GHz,内存 512 MB. 公共就业服务信息网的后台数据库中包含有 385 项就业岗位,在线注册人数共计 124 人(截止至 4 月 26 日 12 时),用户注册时会将各项期望的工作特点填写完毕,包括薪水(1 000 以下、1 000～2 000,2 000～3 000,3 000～4 000,4 000 以上)、职位(主任、高管、副经理等)、地区(北京、上海、济南等)、学历要求(初中、高中、大学、硕士、博士等)以及其他方面,注册界

面如图 4 所示:

图 4 用户注册界面

根据注册用户的信息,所有用户都会填写"月薪范围"这一栏,其他工作特点都存在缺失的情况,因此本实验将"月薪范围"作为用户聚类分组的参照标准,依照薪水高低将所有用户聚类.

4.2 评价指标

用户注册完信息以后,就可以通过系统提供的职位关键词检索寻找理想的职位,第 2 次使用搜索引擎时,系统会通过推荐算法,将用户可能感兴趣的职位直接放置在搜索界面上,如图 5 显示的是第 1 次登录输入的关键字为"Java"的用户 test 第 2 次登录后的界面:

图 5 用户 test 的第二登陆

本实验衡量算法性能的好坏主要依据准确率(precision)的高低,计算公式如下所示:

$$Precision=\frac{\text{第 2 次点击推荐工作的用户数}}{\text{用户总数}}\times 100\%. \tag{11}$$

实验开始,首先比较 KMCURA 和 MPCURA 的性能,本网站投入使用的时间为 1 月 31 日,由于开始时注册用户过少不利于评估,因此从 2 月 20 日至 4 月 20 日共 60 天记录网站的用户数目和准确率计算,以 4 天为单位. 其中 KMCURA 的初始 k 值为 9,随着日期的不断推进,用户数目的不断增加,KMCURA 和 MPCURA 趋势图如图 6 所示.

由图 6 可以看出传统 KMCURA 的准确率上下波动不具备稳定性,而且 k 的取值在第 16 天更新为 23、在第 36 天更新为 41 和在第 48 天更新为 53,无论从维护难度和准确率来看,MPCURA 的效果都优于 KMCURA.

实验第 2 部分:比较 MPCRA、MPCARA 和 MPCURA 的性能. 实验记录时间依旧是从 2 月 20 日开始至 4 月 20 日,并以准确率衡量性能好坏. 三者的趋势图如图 7 所示:

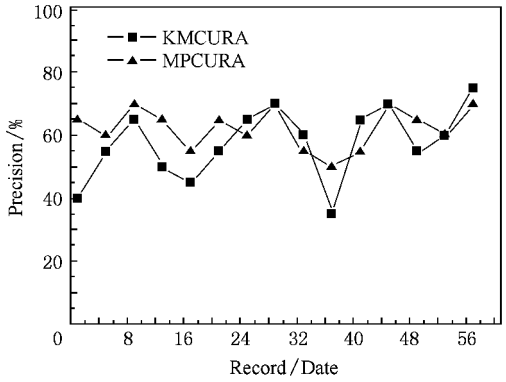

图 6 KMCURA 和 MPCURA 性能比较

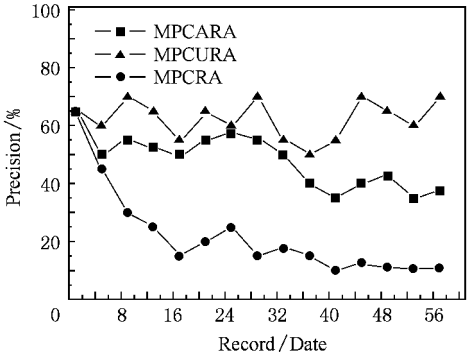

图 7 MPCRA、MPCARA 和 MPCURA 性能比较

由图 7 可以看出,MPCURA 因为具有动态更新的能力可以将推荐系统的效果维持得最好,在准确率方面更能体现出优越性.

4.3 实验结论

通过实验的第 1 部分的比较,可以看出无论是在准确率方面还是在系统维护的难易方面,基于划分聚类算法的最终推荐算法(MPCURA)优于基于传统 k 核心聚类的最终推荐算法(KMCURA)的性能;在实验的第 2 部分,最终推荐算法(MPCURA)的推荐效果是最稳定的,并且具有较高的推荐准确率. 综上所述,本文提出的基于划分聚类的最终推荐算法是最有效的个性化服务推荐算法.

5 结束语

针对 Web 信息处理中数据存在的稀疏性和动态性等问题,本文从分析个性化服务推荐技术的相关原理出发,提出了基于协同过滤的推荐算法和划分聚类算法,并将两者有机结合,更有效地发现用户的兴趣,提高了推荐算法的准确性. 建立了协同过

滤算法的推荐矩阵,并给出评价数值和更新系数,使推荐算法能综合所有用户的评价,为用户提供满意有效的推荐;还提出划分聚类算法,使用户分组更加准确,进一步提高了推荐算法的准确性和实时性,并且在实验中得到验证.

今后的工作主要包括以下3个方面:第一,由于网络信息中的数据十分庞大,如何在大量的非结构文本中抽取符合推荐要求的数据,依然需要十分有效的推荐算法;第二,本文提出的用户聚类,是在具有少数属性的用户基础上完成的,针对高维属性和非数字类属性的用户群体,还要对其做进一步的研究;第三,针对本文已经提出的协同过滤推荐算法和划分聚类算法,我们将继续改进实验方法,不断提高推荐技术的准确性和有效性,更好地满足用户的个性化服务需求.

参 考 文 献

[1] 曾春, 邢春晓, 周立柱. 个性化服务技术综述. 软件学报, 2002, 13(10): 1952-1961

[2] Diaz A, Garcia A, Gervas P. User-centred versus system-centred evaluation of a personalization system. Information Processing and Management, 2008, 44(3): 1293-1307

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

[4] Jiang X, Tan A H. Learning and inferencing in user ontology for personalized Semantic Web search information sciences. Information Science, 2009, 179(16): 2798-2808

[5] Sakkopoulos E, Antonious D, Adamopoulou P, et al. A Web personalizing technique using adaptive data structures: The case of bursts in web visits. The Journal of Systems and Software, 2010, 179(16): 2794-2808

[6] Liu F, Yu C, Meng W. Personalized Web search for improving retrieval effectiveness. IEEE Trans on Knowledge & Data Engineering, 2004, 16(1): 28-40

[7] Aleksandra K M, Boban V, Ivanovic M, et al. E-Learning personalization based on hybrid recommendation strategy and learning style identification. Computers & Education, 2011, 56(3): 855-899

[8] Forsati R, Meybodi M R. Effective page recommendation algorithms based on distributed learning automata and weighted association rules. Expert Systems with Applications, 2010, 37(2): 1316-1330

[9] Kim K J, Cho S B. Personalized mining of web documents using link structures and fuzzy concept networks. Applied Soft Computing, 2007, 7(1): 398-410

[10] 邢春晓, 高凤荣, 战思南, 等. 适应用户兴趣变化的协同过滤推荐算法. 计算机研究与发展, 2007, 44(2): 296-301

[11] 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. 计算机学报, 2010, 33(8): 1437-1455

[12] 黄创光, 印鉴, 汪静, 等. 不确定近邻的协同过滤推荐算法. 计算机学报, 2010, 33(87): 1369-1377

[13] Lee C -H, Kim Y -H, Rhee P -K. Web personalization expert with combining collaborative filtering and association rule mining technique. Expert System with Applications, 2001, 83(11): 2200-2210

[14] Mustapasaa O, Karahocaa D, Karahocaa A, et al. Implementation of semantic Web mining on E-Learning. Procedia Social and Behavioral Sciences, 2010, 2(2): 5820-5823

[15] Chang C C, Chen P L, Chiu F R, et al. Application of neural networks and Kano's method to content recommendation in Web personalization. Expert Systems with Applications, 2009, 36(3): 5310-5316

[16] Alberto DIaz, Antonio GarcIa, Pablo Gervas. User-centred versus system-centred evaluation of a personalization system. Information Processing and Management, 2008, 44(3): 1293-1307

[17] 吴湖, 王永吉, 王哲, 等. 两阶段联合聚类协同过滤算法. 软件学报, 2010, 21(5): 1042-1054

吴泓辰 男,1988年生,硕士研究生,主要研究方向为数据库与信息检索.

王新军 男,1968年生,博士,教授,博士生导师,主要研究方向为数据集成、XML数据管理、数据库与信息检索.

成 勇 男,1977年生,博士,高级工程师,主要研究方向为数据库、信息检索.

彭朝晖 男,1978年生,博士,副教授,硕士生导师,主要研究方向为数据库与信息检索.