

基于 PH-Tree 多属性索引树的朋友推荐算法

梁俊杰 孙阳征

(湖北大学计算机与信息工程学院 武汉 430062)

摘 要 现阶段越来越多的人通过社交网络结识新朋友,如何为用户快速准确推荐朋友是考量平台服务能力的重要指标。提出了一种基于索引树的在线网络朋友推荐方法,即基于网络结构局部特性的思想将用户间的多属性相交值转换为二进制位码向量表示,利用改进后的 PH-Tree 将所有的向量构造成排序索引树,通过遍历索引树容易确定用户的最佳推荐朋友集。实验证明本方法具有较高的效率和准确率。

关键词 社交网络,PH-Tree,索引,朋友推荐

中图法分类号 TP311.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.4.031

Friend Recommendation Algorithm Based on PH-Tree Multi-attribute Index Tree

LIANG Jun-jie SUN Yang-zheng

(School of Computer and Information Engineering, Hubei University, Wuhan 430062, China)

Abstract Nowadays, more and more people make new friends on online social networks (OSN), and it becomes an important feature for OSN to recommend friends for users rapidly and exactly. An online friend recommended algorithm was proposed using a sorted index tree based on local network structure. Multi-attribute intersection values between different users were converted into binary vectors and organized in PH-Tree. Thus a user's best friend recommended set can be determined by travelling the index tree easily. The experiments show that our method performs efficiently and precisely.

Keywords Social networks, PH-Tree, Index, Friend recommendation

1 引言

随着下一代互联网技术的快速发展,在线社交网络 OSN (Online Social Networks)吸引了大量用户的关注和应用,如 Facebook、Linkedin 和人人网等成为了主流的互动网络平台。当前在线社交网络平台流行的主要原因是用户通过平台不仅可以与其他用户进行互动交流和订阅自己感兴趣的信息,还可以结识更多的新朋友,因此平台能否为用户快速准确地推荐朋友已成为衡量 OSN 平台技术的重要指标^[1]。

由于社交平台用户需求和模式的不同,现有的平台采用了不同的方法为用户提供朋友推荐服务,这些方法基本上可以分为 2 种:基于网络结构的局部特性和基于网络结构的全局特性。如 Face book 采用的是将用户朋友的朋友 FOAF^[2] (Friend of a Friend)推荐给用户,它是根据每个候选朋友和目标用户的共同朋友的数目来推荐好友。作为基于网络结构的局部特性推荐朋友的典型算法,FOAF 在聚集系数较高的网络中能够产生较好的推荐效果,但是存在对 OSN 信息利用率不够高的问题^[3]。基于网络结构的全局特性,如 Google 网页排序算法 Page Rank^[4],该推荐方法能侦测网络的所有路径结构,推荐的准确率相对局部网络结构更高,但是

随着 OSN 中用户爆炸式的增长,基于全局特性进行推荐的效率将受到很大影响。

为此,针对目前基于网络结构局部特性进行朋友推荐方法的准确率和效率较低问题,本文提出基于 PH-Tree [5]多属性索引树的朋友推荐算法 RMPH-Tree (Recommended in Multi-attribute PH- Tree),即在目标用户的朋友圈中将候选对象与目标用户的多个属性之间的相似值表示为位码向量,构造 PH-Tree 排序树来实现准确高效的朋友推荐过程。通过大量实验证明 RMPH-Tree 算法与现有方法相比在推荐准确性和性能方面表现更好。

2 相关工作

2.1 背景知识

在社交平台中用户与用户之间的朋友关系称为朋友圈。朋友圈可以用无向图 $G=(U, E)$ 来表示^[6],其中 U 表示用户, E 表示用户与用户之间的朋友关系。20 世纪 60 年代,美国著名社会心理学家 Milgram 提出“小世界”理论,即世界上的每个人只需要通过平均 6 个人(最多不超过 6 个人)就可以和任意一个人建立起联系。根据该理论,RM PH-Tree 将朋友圈的推荐路径长度 L 限定在 $2 \leq L \leq 6$ 范围内。

收稿日期:2014-05-18 返修日期:2014-08-30 本文受国家发改委 2012 年信息安全专项(发改办高技[2013]1309 号),湖北省自然科学基金重点项目(2013CFA115),武汉市科技攻关计划项目(2013012401010851)资助。

梁俊杰(1974-),女,副教授,主要研究方向为大数据、数据库、云计算,E-mail:416356541@qq.com;孙阳征(1991-),男,硕士生,主要研究方向为大数据、数据分析,E-mail:327708519@qq.com。

文中所有符号的说明如表 1 所列。

符号	具体含义
$U(U_1, U_2, \dots, U_i)$	用户集
$\Phi(\Phi_1, \Phi_2, \dots, \Phi_s)$	用户属性集合
$C(c_1, c_2, \dots, c_s)$	相交属性值向量
$W(w_1, w_2, \dots, w_s)$	相交属性权重向量
L	推荐路径长度
K	前 K 个推荐朋友
M	用户的候选朋友个数

2.2 PH-Tree 简介

PH-Tree 是一种基于向量位码 (bitcode) 表示的索引树^[7], 通过共享位码前缀而节省存储空间^[8], 实现查询时效率的提高。例如, 假设二维空间中的 3 个点 $P_1(7, 8)$ 、 $P_2(5, 8)$ 和 $P_3(7, 10)$, 对应的位码分别为 $P_1(0111, 1000)$ 、 $P_2(0101, 1000)$ 和 $P_3(0111, 1010)$, 通过观察可以看出, 这 3 个点的前两位和最后一位位码是相同的 (分别是 01、10 和 10), 因此在 PH-Tree 中这 3 层的位码数据只存储一次, 如图 1 所示。

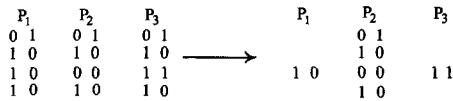


图 1 PH-Tree 索引树示意图

3 RMPH-Tree 索引树构造

3.1 多属性表示方法

现实生活中相同点越多的人越容易成为朋友^[9], 因此我们可以利用 OSN 提供的用户各项信息, 如兴趣、性别、年龄、工作经历、学习经历等^[10], 综合分析用户的可能朋友集合并进行推荐, 以此提高推荐的准确性。其中, OSN 提供的用户每项信息都可以看作是一个用户属性, 则用户可以表示为多个属性的集合 $U(\Phi_1, \Phi_2, \dots, \Phi_s)$, s 代表属性个数。

例如, 假设用户 U_1 的两个属性分别为朋友和兴趣, 其中朋友 $\Phi_{11} = \{张三, 李四, 何里, 李燕, 于爽, 马腾, 王昕, 钱秋, 何水波, 陈胜, 吴波, 熊参\}$, 兴趣 $\Phi_{12} = \{羽毛球, 旅游, 篮球, 足球\}$, 则用户 U_1 可以表示为 $U_1(\Phi_{11}, \Phi_{12})$ 。

为了向用户准确地推荐朋友, 需要综合考虑推荐对象与被推荐对象的多项属性, 两者之间各个属性集间存在的相同属性值越多, 则成为朋友的可能性就越大^[11]。

定义 1(相交属性值) 对于属性 s , 用户 U_i 与 U_j 具有的共同属性值个数, 记为 $c_s(U_i, U_j)$ 。

例如用户 U_1 的朋友 $\Phi_{11} = \{张三, 李四, 何里, 李燕, 于爽, 马腾, 王昕, 钱秋, 何水波, 陈胜, 吴波, 熊参\}$, 用户 U_2 的朋友 $\Phi_{21} = \{张三, 李四, 于发\}$, 则用户 U_1 和用户 U_2 在朋友属性上的相交属性值 $c_1(U_1, U_2) = 2$ 。

定义 2(相交属性向量) 用户 U_i 和用户 U_j 在所有属性上的相交属性值组成的向量, 记为 $C(U_i, U_j) = (c_1, c_2, \dots, c_s)$, 其中 $c_i (1 \leq i \leq s)$ 是第 i 个相交属性值。

例如假设 U_1 和 U_{14} 有 5 个共同朋友和 1 个共同兴趣 (其中相同朋友个数为第一属性, 相同兴趣数为第二属性), 则表示为 $C(U_1, U_{14}) = (5, 1)$ 。

实际应用时, 考虑到 OSN 可能对不同的属性在朋友推荐中起到的影响力不同, 系统对不同的属性需要赋予不同的权重值^[12], 因此在构建 PH-Tree 之前必须对相交属性值根据权重从大到小进行排序。

定理 1(相似值) 假设 $C(U_i, U_j) = (c_1, c_2, \dots, c_s)$ 中对应的权重向量为 $W(w_1, w_2, \dots, w_s)$, 根据权重从大到小排序后的属性向量为 $C(U_i, U_j) = (c_1', c_2', \dots, c_s')$, 对应的权重向量为 $W'(w_1', w_2', \dots, w_s')$, 则 U_i 和 U_j 间的相似值可以用公式: $Sim_attribute = \sqrt{w_1'(c_1')^2 + w_2'(c_2')^2 + \dots + w_s'(c_s')^2}$ 来度量。

由定理 1 可知, 两个用户之间的相似值越大, 那么这两个用户越相似, 越值得相互推荐成为朋友。为利于相交属性向量的使用和管理, 本文在 PH-Tree 中为相交属性加入权重, 使树中每层右边的值大于左边的值, 从而成为一种新的排序 PH-Tree, 即 RMPH-Tree, 其能够更准确快速地为用户推荐朋友。

3.2 RMPH-Tree 多属性树构造

在采用 PH-Tree 组织用户属性之前, 需要将相交属性向量转换成位码向量表示形式。转换表示时根据相交属性向量中的最大值确定位码表示的二进制长度, 即 PH-Tree 存储数据的位码长度。

定义 3(相交属性位码向量) 将十进制表示的相交属性向量转化为二进制表示, 即为相交属性位码向量。

例如: $C(U_1, U_{14}) = (5, 1)$ 转化成 4 位二进制表示为 $C(U_1, U_{14}) = (0101, 0001)$ 。

为描述方便, 假设用户多属性包括朋友、兴趣和年龄^[13], 被推荐对象为 U_1 , 则已经和 U_1 是朋友的用户将不再考虑, 根据朋友圈和推荐路径长度限制, 需要确定 U_{14} 、 U_{15} 、 U_{16} 和 U_{17} 推荐给 U_1 做朋友的先后顺序, 这一难题可以利用 RMPH-Tree 来解决。

RMPH-Tree 是将用户的相交属性位码向量组织成 PH-Tree 的索引结构。RMPH-Tree 具体构造过程以实例来说明。

(1) 根据朋友圈图 (见图 2), 确定 U_1 分别与 U_{14} 、 U_{15} 、 U_{16} 、 U_{17} 在第一个属性“朋友”上的相交属性值 $c_1(U_1, U_{14}) = 5$, $c_1(U_1, U_{15}) = 0$, $c_1(U_1, U_{16}) = 0$, $c_1(U_1, U_{17}) = 7$ 。

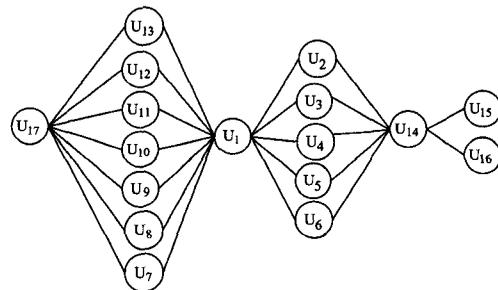


图 2 U_1 的朋友圈关系

(2) 根据 OSN 提供的用户兴趣信息 (见图 3), 确定 U_1 分别与 U_{14} 、 U_{15} 、 U_{16} 、 U_{17} 在第二个属性“兴趣”上的相交属性值, $c_2(U_1, U_{14}) = 1$, $c_2(U_1, U_{15}) = 1$, $c_2(U_1, U_{16}) = 0$, $c_2(U_1, U_{17}) = 2$ 。

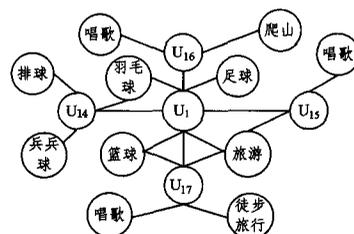


图 3 用户兴趣关系

(3)根据 OSN 提供的用户年龄数据以及交友年龄范围信息(见表 2),交友的年龄范围里的每个值可看作是一个点,由此确定 U_1 与每个候选用户在第三个属性“年龄”上的相交属性值, $c_3(U_1, U_{14}) = 1, c_3(U_1, U_{15}) = 2, c_3(U_1, U_{16}) = 2, c_3(U_1, U_{17}) = 0$ 。

表 2 用户年龄范围表

用户	年龄	交友的年龄范围
U_1	20	[18,22]
U_{14}	16	[14,18]
U_{15}	17	[15,19]
U_{16}	23	[21,25]
U_{17}	28	[26,30]

(4)将上述 3 个步骤的向量合并得到 U_1 与 $U_{14}, U_{15}, U_{16}, U_{17}$ 的相交属性向量 $C(U_1, U_{14}) = (5, 1, 1), C(U_1, U_{15}) = (0, 1, 2), C(U_1, U_{16}) = (0, 0, 2), C(U_1, U_{17}) = (7, 2, 0)$ 。

(5)根据系统给出的权重对相交属性向量进行排序,假设 3 个属性的权重从大到小依次为:朋友、年龄、兴趣,则排序后的相交属性向量为 $C(U_1, U_{14}) = (5, 1, 1), C(U_1, U_{15}) = (0, 2, 1), C(U_1, U_{16}) = (0, 2, 0), C(U_1, U_{17}) = (7, 0, 2)$ 。

(6)用二进制表示法将相交属性向量表示为相交属性位码向量为: $C'(U_1, U_{14}) = (0101, 0001, 0001), C'(U_1, U_{15}) = (0000, 0010, 0001), C'(U_1, U_{16}) = (0000, 0010, 0000), C'(U_1, U_{17}) = (0111, 0000, 0010)$ 。

(7)利用 PH-Tree 将所有的相交属性位码向量组织起来,同时在树的同层数据中 RMPH-Tree 按照数值从小到大排序,这样在 RMPH-Tree 遍历时从根节点到叶子节点的路径来看,右路节点数据要大于左路节点数据,因此利用 PH-Tree 可以实现对向量数据的排序组织,即 RMPH-Tree 为朋友推荐排序索引树(见图 4)。

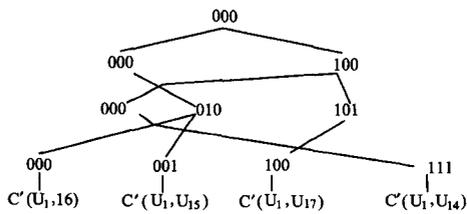


图 4 RMPH-Tree 排序索引树

根据 RMPH-Tree 的构造过程可知,在 RMPH-Tree 的叶子节点中,右端节点与用户 U_1 的相似度比左端的要大;并且从根到叶子节点的路径来看,右线比左线的相似性度量值大,因此利用 RMPH-Tree 的深度遍历和从右至左遍历即可得到为 U_1 推荐朋友的最佳顺序为: $U_{17}, U_{14}, U_{15}, U_{16}$ 。

4 基于 RMPH-Tree 的朋友推荐算法

基于 RMPH-Tree 的朋友推荐算法如下:

输入:用户集合 U ,属性权重向量 W ,被推荐用户 U_1

输出:最佳推荐朋友排序集 SC 。

步骤:

- (1)初始化推荐朋友 $SC = \emptyset$;
- (2)在朋友圈图 $G = (U, E)$ 中将与用户 U_1 的路径长度大于 2 且小于等于 6 的用户加入候选对象集合 $RS_temp = (U_k, U_w, \dots, U_1)$;
- (3)确定 U_1 与 RS_temp 中的每个候选对象的相交属性值 $C(U_1, U_i)$,其中 $U_i \in RS_temp$;
- (4)按权重从大到小对相交属性值中的各个向量进行排序,并转化为

相交属性位码向量;

(5)构建 RMPH-Tree;

(6)结合深度遍历和从右至左的广度遍历将 RMPH-Tree 叶子节点依次添加到 SC 中;

(7)返回 SC 。

5 实验验证和结果分析

5.1 实验数据

该算法的验证数据由美国东北大学的 Alan Mislove^[14] (<http://socialnetworks.Mpi-sws.org>)提供,数据采集了在新奥尔良地区的 Facebook 朋友关系,有 63578 个注册用户,包括用户的朋友数、兴趣爱好和年龄等属性。本实验随机选取 10 个朋友圈包括 1000 个用户信息来对该算法进行验证。

实验代码用 MyEclipse 10.0 编写,Java 语言实现,jdk 版本为 1.7。实验代码运行于 MacBook Pro ME865CH/A,操作系统 Mac OS X Mavericks,CPU 2.4GHz 双核 Intel Core i5,8GB 1600MHz DDR3L 内存,256G 固态硬盘。

5.2 评价标准

RMPH-Tree 是基于网络结构局部特性的朋友推荐算法,实验选取同类型的协同过滤算法^[15](Collaborative Filtering, CF)和朋友的推荐算法(FOAF)作为比较对象,验证 RMPH-Tree 算法的先进性。实验中根据准确率^[16](Precision@k)、召回率(Recall)^[15]和平均准确率(Mean Average Precision, MAP)^[15]3 个指标来评价推荐算法的准确性。

准确率(Precision@K)表示正确推荐的朋友数目占所有推荐朋友数目的比例。P@K 表示前 K 个推荐朋友中正确推荐所占的比例,记为 $P@K = \text{正确推荐朋友数}/K$ 。召回率(Recall)表示正确推荐朋友个数与测试集中总的候选朋友个数 M 的比例,记为 $\text{Recall} = \text{正确推荐朋友数}/M$ 。平均准确率(MAP)表示对所有用户推荐结果的准确率的平均值,记为 $\text{MAP} = \sum_{k=1}^K AP(k)/t$,该值越大说明算法的准确率越高。MAP 从整体上评估算法的准确性,AP 表示对每个用户推荐结果的平均准确率。

5.3 实验

5.3.1 属性个数对准确率的影响

多属性是 RMPH-Tree 算法实现准确推荐的重要因素,为了验证多属性能提高算法的准确性,在实验中将测试属性数量的变化对推荐准确率的影响。假设为用户推荐朋友个数为 K ,选取的属性个数为 S ,实验中分别测试当 K 为 20、30,测试 S 取值 5、10 和 15 时的推荐结果准确率。为了保障实验的有效性,我们将对每组数据做 5 次测试,然后取平均值作为测试结果,如表 3 所列。

表 3 属性个数对准确率的影响

K	属性个数(S)	准确率(P@K)
K=20	5	20.9%
	10	26.4%
	15	33.7%
K=30	5	18.3%
	10	25.1%
	15	31.9%

由实验知,随着推荐个数 K 的增大,准确率将有小幅下降。这是因为 RMPH-Tree 是依据深度遍历和从右至左遍历依次产生推荐朋友顺序,随着推荐人数的增多,推荐的朋友相

似性将越来越小,因此准确率随推荐人数的增多而降低。在 K 值相同时,随着选取属性个数的增多,准确率明显提高,这是因为属性越多,用户之间的相似性度量考虑越周全,越能够做出准确的推荐。

由此可见,属性个数对提高算法的准确性起到至关重要的作用。为实验方便,以下将选择属性数量 $S=15$ 进行测试。

5.3.2 数据规模对运行时间的影响

通过选取 3 种不同的测试集数据规模,比较了 3 种算法在不同规模数据集下的运行时间,实验结果如图 5 所示。分析可知,随着数据规模的增大,3 种算法的运行时间都呈增长趋势。但是,相比 CF 和 FOAF 而言,RMPH-Tree 算法的运行时间最短,只是 CF 算法的 20%和 FOAF 的 55%。这是因为 RMPH-Tree 算法采用 PH-Tree 索引结构来存储数据,通过共用前缀的存储形式大大减少了存储空间,从而降低了 I/O 代价。因此 RMPH-Tree 算法能够更快地处理数据,具有更好的效率和稳定性。

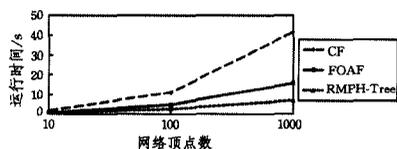


图 5 3 种算法运行时间的比较

为了评估 RMPH-Tree 的准确性推荐效果,实验从准确率、召回率和平均准确率 3 个方面对 RMPH-Tree 算法、FOAF 算法和 CF 算法进行比较。

5.3.3 准确率

实验选取前 K 个推荐朋友的数量分别为 20、30、40 和 50,观察 3 种算法的准确率变化情况(见图 6),可以看出 RMPH-Tree 算法的准确率一直高于 FOAF 算法和 CF 算法;同时随 K 值的增大,FOAF 算法和 CF 算法的准确率下降迅速,而 RMPH-Tree 算法的准确率一直保持在 0.35 的水平。这是由于 RMPH-Tree 算法采用排序后的 PH-Tree 来存储数据,在推荐时不仅能够快速地判断出最佳朋友,而且 PH-Tree 能够依据朋友相关性由强至弱进行推荐,而 FOAF 算法和 CF 算法没有采用索引结构,并且没有考虑其他用户属性,仅仅根据局部网络结构中顶点相似性不能做到 RMPH-Tree 程度的准确推荐。因此 RMPH-Tree 算法有更高的推荐准确率。

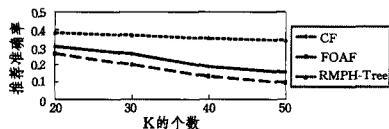


图 6 在不同数据规模下 3 种算法的 Precision 值

5.3.4 召回率

如图 7 所示,实验随机选取 5 个用户,候选朋友个数分别为:39,46,32,51,29,实验结果取平均值,测试在不同测试集规模情况下的召回率。由于测试集包含了与被推荐用户无关的用户,因此召回率明显低于准确率。由实验可以看出 RMPH-Tree 算法的召回率始终高于 FOAF 算法和 CF 算法的,分析可知,RMPH-Tree 算法在构建 PH-Tree 时,就已经决定了朋友推荐结果顺序,且不会因为新加进来的无关用户破坏 RMPH-Tree 的结构。这是由于 RMPH-Tree 是根据用户属

性值而构建,无关用户与被推荐用户之间的相交属性值很小甚至为 0,这样在构建 RMPH-Tree 时无关用户排在树的最左端或者根本不出现在 RMPH-Tree 中。因此,RMPH-Tree 算法能够保持较高的召回率。

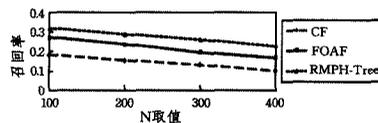


图 7 不同测试集规模下的召回率

5.3.5 平均准确率

如图 8 所示,根据以上实验可知 RMPH-Tree 的准确率明显高于其它两种算法,不难看出 MAP 从整体上说明了 RMPH-Tree 算法更好。

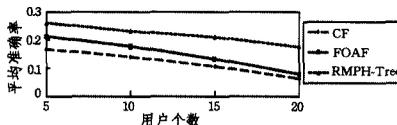


图 8 平均推荐准确率 MAP 值

以上实验结果综合表明,RMPH-Tree 算法能够更快更准确地进行朋友推荐,当候选朋友数很少且属性信息较少时,本文算法的推荐准确率也会大幅下降^[17],甚至不及其余的一些推荐算法,因为本文算法主要是对属性值进行分析的。

结束语 向用户推荐新朋友是社交网络平台的重要功能之一,如何快速准确地向用户推荐朋友是衡量平台服务能力的重要指标^[18]。本文创新地将索引结构和推荐算法相结合,提出了一种新的朋友推荐排序索引树 RMPH-Tree,即通过将用户间的多属性相交值转换为二进制位码表示并组织成 PH-Tree 结构,利用树遍历过程来确定朋友推荐的最优顺序集^[19]。实验证明 RMPH-Tree 相比同类算法具有更高的准确率和效率。

在接下来的工作中,我们将更多地考虑社交网络平台的用户实时数据^[20]以及属性细分对推荐准确性和效率的影响。

参考文献

- [1] Yin Z, Gupta M, Weninger T. A unified framework for link recommendation with user attributes and graph structure [C]// Proceedings of the 19th International Conference on World Wide Web. New York, USA, 2010, 6:1200-1212
- [2] Chen J, et al. Make new friends, but keep the old: Recommending people on social networking sites [C]// Proceeding of the 27th International Conference on Human Factors in Computing Systems. New York; ACM, 2009, 12:201-210
- [3] 付艳艳,张敏,冯登国,等.基于节点分割的社交网络属性隐私保护[J].软件学报,2014,25(4):768-780
- [4] 余琰,邱广华.基于局部随机游走的在线社交网络朋友推荐算法[J].系统工程,2013,31(2):47-54
- [5] Zaslachke T, Zimmerli C, Norrie M C. The PH-Tree: A Space-Efficient Storage Structure and MultiDimensional Index. Institute for information Systems [J]. Department of Computer Science ETH Zurich, Switzerland, 2014, 7:659-671
- [6] Wolfe A W. Social network analysis: Methods and applications [J]. Cambridge American Ethnologist, 1997, 24(1):210-230

(下转第 176 页)

和精确度。可以看出,RLOF 算法在处理低维数据时执行速度较快;但随着维度增加,基于 R2-tree 索引的查询成本增加,基于子空间划分的检测算法更占优势。RLOF 算法在检测精度上效果更好。

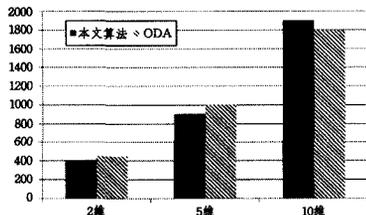


图 4 执行效率比较

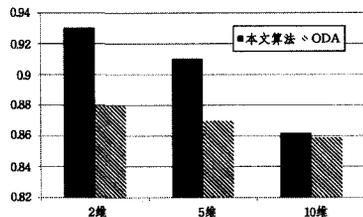


图 5 精确度比较

结束语 不确定数据的离群点检测是一个非常重要的研究领域,具有广泛的应用前景。本文提出了一种基于密度的局部离群点检测算法 RLOF。该方法通过引入 R2-tree 这种数据结构,在最小充分邻域的确定和数据集动态更新环节上缩短了计算时间,提高了整体算法的执行效率。实验表明,该算法一定程度上能够适应空间和海量数据环境下的离群点检测。下一步,将充分考虑不确定数据的属性级不确定性对算法的影响和面向不确定数据流的离群点检测算法。

参考文献

[1] Breunig M M, Kriegel H P, Ng R T, et al. LOF: identifying density-based local outliers[J]. ACM Sigmod Record, 2000, 29(2): 93-104

[2] Tu L, Cui P, Tang K. A Density Grid-Based Clustering Algorithm for Uncertain Data Streams[C]//2013 10th Web Information System and Application Conference (WISA). IEEE, 2013;

347-350

[3] Chawla S, Gionis A. k-means: A Unified Approach to Clustering and Outlier Detection[C]//SDM. 2013;189-197

[4] Duforet-Frebourg N, Blum M G B. Bayesian Matrix Factorization for Outlier Detection: An Application in Population Genetics[M]//The Contribution of Young Researchers to Bayesian Statistics. Springer International Publishing, 2014;143-147

[5] Cao K, Han D, Wang G, et al. An Algorithm for Outlier Detection on Uncertain Data Stream[M]//Web Technologies and Applications. Springer Berlin Heidelberg, 2013;449-460

[6] Aggarwal C C, Philip S Y. Outlier Detection with Uncertain Data[C]//SDM. 2008;483-493

[7] Yang C, Lin K I. An index structure for efficient reverse nearest neighbor queries[C]//17th International Conference on Data Engineering, 2001. IEEE, 2001;485-492

[8] Cao K, Han D, Wang G, et al. An Algorithm for Outlier Detection on Uncertain Data Stream[M]//Web Technologies and Applications. Springer Berlin Heidelberg, 2013;449-460

[9] Hjaltason G R, Samet H. Distance browsing in spatial databases [J]. ACM Transactions on Database Systems (TODS), 1999, 24(2):265-318

[10] Aggarwal C C. On density based transforms for uncertain data mining[C]//IEEE 23rd International Conference on Data Engineering, 2007(ICDE 2007). IEEE, 2007;866-875

[11] HU Cai-ping, QIN Xiao-lin. A Density-Based Local Outlier Detecting Algorithm[J]. Journal of Computer Research and Development, 2010(12);2110-2116

[12] Zhou A Y, Jin C Q, Wang G R, et al. A survey on the management of uncertain data[J]. Chinese Journal of Computers, 2009, 32(1);1-16

[13] Yu Hao, Wang Bin, Xiao Gang, et al. Distance-Based Outlier Detection on Uncertain Data[J]. Journal of Computer Research and Development, 2010, 47(3);474-484

[14] Guttman A. R-trees: A dynamic index structure for spatial searching[M]. ACM, 1984

[15] Yu Min-min, Cheng Ning-jiang. Algorithm of Improved Top-k Query on Uncertain Data for Requirement Extension[J]. Computer Science, 2012, 39(6);151-154

(上接第 159 页)

[7] 印鉴,王智圣,李琪,等. 基于大规模模式反馈的个性化推荐[J]. 软件学报, 2014, 25(9);1953-1966

[8] German U, Joanis E, Larkin S. Tightly Packed Tries: How to Fit Large Models into Memory, and Make them Load Fast, Too[C]//Proc. of the NAACLHLT Workshop. 2009, 8;31-39

[9] Leskovec J, Lang K J, Mahoney M. Empirical comparison of algorithms for network community detection [C]//Proceedings of the 19th International Conference on World Wide Web. New York; ACM, 2010, 19;630-655

[10] 徐志明,李栋,刘挺,等. 微博用户的相似性度量及其应用[J]. 计算机学报, 2014(1);207-218

[11] 贾大文,曾承,彭智勇,等. 一种基于用户偏好自动分类的社会媒体共享和推荐方法[J]. 计算机学报, 2012(11);2381-2391

[12] 张丰,王箭,赵燕飞,等. 社交网络中一种基于社区推荐的信任模型[J]. 计算机科学, 2014, 41(5);168-172

[13] 王玛,高琳. 基于社交圈的在线社交网络朋友推荐算法[J]. 计算机学报, 2014(4);801-808

[14] Alan M, Massimiliano M, Krishnap G, et al. Bhattacharjee Bobby, Measurement and analysis of online social network[C]//Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement. San Diego, CA, USA, 2007, 5;29-45

[15] Chen Ke-han, Han Pan-pan, Wu Jian. User Clustering Based Social Network Recommendation[J]. College of Computer Science and Technology, 2013, 3;100-107

[16] Herlocker J. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information systems, 2004, 22(1);5-53

[17] 文俊浩,舒珊. 一种改进相似性度量的协同过滤推荐算法[J]. 计算机科学, 2014, 41(5);68-71

[18] 高明,金澈清,钱卫宁,等. 面向微博系统的实时个性化推荐[J]. 计算机学报, 2014(4);963-975

[19] 海本高,解瑞云. 基于贝叶斯网络的上下文推荐算法[J]. 计算机科学, 2014, 41(7);275-278

[20] 朱宝,徐玲玉. 一种个性化推荐方法[J]. 计算机科学, 2014, 41(11A);294-297