基于最小圆覆盖区域划分的索引过滤算法

陈 洁[1),2)] 方滨兴[1),2)] 谭建龙[2)] 金世超[2),3)]

[1)](北京邮电大学计算机学院 北京 100876)

[2)](中国科学院信息工程研究所信息智能处理实验室 北京 100093)

[3)](北京大学软件与微电子学院 北京 100084)

摘 要 过滤算法设计是信息内容安全处理系统中的一个重要环节,过滤速度成为衡量过滤系统性能的首要因素.索引结构是处理大规模数据的一种有效方式,但目前索引方法都是针对特定检索领域而设计,在实际过滤应用中,并不能满足过滤实时性需求.为了加快信息过滤中数据查询的判定速度,文中提出一种基于最小圆覆盖的区域划分方法,构建了适合过滤的索引结构:F-tree.该算法充分考虑实际过滤环境中正例(正常信息)多、反例(敏感信息)少的非平衡数据分布特性,利用最小圆覆盖划分方法得到最大否定判断区域.在查询阶段,正例以最大概率落入否定区域,根据否定性判定原理可以对正例快速否定判定,从而加快整体查询的判定速度.实验表明,与现有算法相比,所提出的算法减少了查询中的距离计算次数,有效提高了过滤查询性能.

关键词 过滤算法;最小圆覆盖;否定性判定;索引结构

中图法分类号 TP391 **DOI号**: 10.3724/SP.J.1016.2012.02139

Index Filtering Algorithm Based on Minimum Enclosing Circle Partition

CHEN Jie[1),2)] FANG Bin-Xing[1),2)] TAN Jian-Long[2)] JIN Shi-Chao[2),3)]

[1)](*School of Computer Science, Beijing University Posts and Telecommunications, Beijing* 100876)

[2)](*National Engineering Laboratory for Information Security Technologies, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)

[3)](*School of Software and Microelectronics, Peking University, Beijing* 100084)

Abstract Filtering algorithm design play a very important role in information content security process system, filtering speed become the first consideration factor for improving the system performance. Index is an effective method to cope with massive data and can get a good performance. Unfortunately, most of existing index methods especially designed for information retrieval application and these indexes cannot achieve a good performance for filtering application. In order to improve the filtering performance, this paper proposes an index filtering method based on minimum enclosing circle data partition, and built a particular filtering index called F-tree. This method considers the imbalance data distribution with more positive and less negative in real filtering situation, the minimum enclosing circle partition method is used to obtain maximum negative area. In query step, the positive data falls into negative area with maximum probability in order to get up to the all data judgment speed. Experimental results show that the proposed method can improve the filtering performance by reducing the times of computation.

Keywords filtering method; minimum enclosing circle; negative judgment; index structure

---

收稿日期:2012-06-30;最终修改稿收到日期:2012-08-23.本课题得到国家"八六三"高技术研究发展计划项目基金(2011AA010705,2012AA012502)资助.**陈 洁**,女,1982年生,博士研究生,主要研究方向为信息安全、相似性搜索、索引结构.E-mail:chenxg58@126.com.**方滨兴**,男,1960年生,博士,教授,博士生导师,主要研究领域为信息安全等.**谭建龙**,男,1974年生,博士,教授,博士生导师,主要研究领域为信息安全、相似性搜索、索引技术.**金世超**,男,1988年生,硕士研究生,主要研究方向为索引结构、相似性搜索.

1 引 言

信息过滤是信息内容安全领域具有实际应用价值的一个研究方向.针对网络中存在的敏感信息,设计有效的过滤算法,对用户所请求的网页信息进行审计,从而过滤判定为敏感内容的信息.过滤算法设计是整个安全过滤系统中的一个重要环节,决定过滤系统整体的速度性能.一般认为敏感信息过滤技术实质上是一个二分类的问题,目前过滤算法主要采用基于模式分类的方法,通过学习基本的模型,为每个测试数据都打上一个标签,从而过滤标记为敏感信息的数据,常用的有基于概率论的贝叶斯分类方法以及基于最大间隔的 SVM 分类方法.而在实际过滤应用中我们需要面对的是海量数据的处理,数据量的个数远远超出了传统模式分类方法的应对能力.因此为特征集建立有效的索引结构是实现大规模数据高性能检测的关键技术.目前在信息检索领域已有大量有效的索引构建方法被提出,虽然信息检索和信息过滤在某些关键技术的运用上有相似之处,但信息安全过滤相比信息检索需要更快的判定速度,在数据处理方式上也不同.目前基于检索的索引结构根本不能满足实际过滤处理环境中所需的实时性需求.

本文在分析实际网络中正例与反例非平衡分布特性的基础上,提出了一种面向安全过滤的基于最小圆覆盖区域划分的索引构建方法.该方法利用否定性判定的思想,结合最小圆覆盖理论基础,通过对数据进行最小圆划分,使每个最小圆之间呈现稀疏性分布特性,以此构建适合实际网页过滤的高性能索引结构 F-tree.本文的主要贡献包括:

(1)深入分析敏感信息过滤的特性,引入图形学中最小圆覆盖理论来解决实际的应用问题.

(2)提出一种新的区域划分策略——最小圆覆盖区域划分,并基于此划分构建了适合快速过滤的索引结构 F-tree,以满足过滤实时性需求.

(3)在大量数据集上,对文本提出的方法进行了实验验证.

多组实验结果证明,所提出的过滤索引结构在查询中所需的距离比较次数在不同搜索半径条件设置下都少于经典的 M-tree 检索结构,在过滤应用中表现出极高的查询性能.

本文第 2 节分析过滤索引树结构特点;第 3 节描述所需要解决的问题;第 4 节介绍基于最小圆区域划分的过滤索引结构 F-tree 的构建方法;第 5 节通过实验验证所提出算法的有效性;最后是结束语.

2 过滤索引树结构特点

相似性索引方法一般应用在检索领域,是一种处理大规模数据或高维数据的有效技术手段,其基本原理是根据数据的分布特性,构建有效的索引结构组织数据,从而加快数据检索的速度.典型的检索索引结构可以简单地用图 1 来表示,检索的主要特点是尽量多地检索出相关的数据.因此,有效的检索树结构要近似平衡树,一般是搜索到叶子节点才返回结果,这个结果可以直接反馈给用户,让用户评价,从而更新用户模板,使搜索结果更精确.因此,在构建基于检索的索引结构时,主要考虑尽量多的搜索出满足用户所需求的数据.

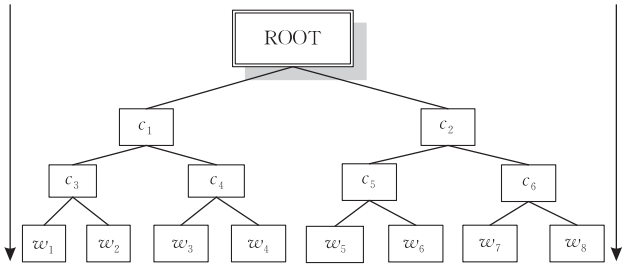


图 1 检索树结构

由于安全过滤具有实时性和否定判断的特性,这个特性与基于检索的索引结构在本质上有所不同,因此适合过滤的索引结构和适合检索索引结构有很大区别.当构建一个适合过滤的索引结构后,给定查询 q,我们不需要找出与它相似的多个数据对象,因为在安全过滤中没有反馈,不需要把结果反馈给用户,只需要尽快判定它的特性,从而直接进行处理.因此,我们构建的过滤索引结构,在查询阶段不需要每个查询都搜索到叶子节点才返回结果,而是查询到部分中间节点时,就可以判定查询数据的特征,从而退出查询.典型的过滤索引结构可以用图 2 简单描述.在查询阶段查询数据不需要遍历到

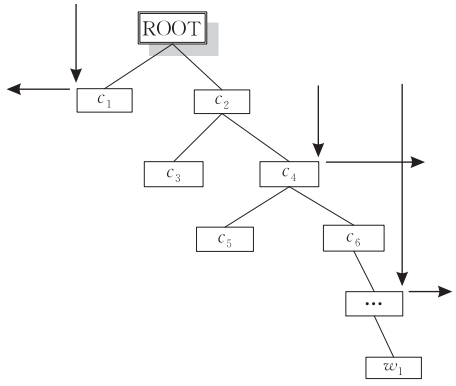


图 2 过滤树结构

叶子节点，只需要使判定结果尽可能在树的中间节点就返回. 通过设计中间节点的返回模式，加快过滤应用中数据判定速度.

3　问题描述

根据第 2 节的分析可知，具有高性能特性的过滤索引树需要在中间节点就返回结果. 根据以上特点，在构建索引结构过程中，数据空间的分割是否合理，会直接影响索引结构的性能. 目前存在的区域划分方法（如基于支点选择的方法[1-2]，还有基于划分的方法[3]），大部分都是遵循层次化聚类的原则. 每一种索引都有自己的特性，包括数据类型、数据分布情况还有维度大小. 这些特性对索引结构的性能都有较大的影响. M-tree 索引[4] 至今被认为是性能最优的索引结构之一，这种索引结构是根据数据空间中各个数据点之间的距离进行空间划分的，将一个数据空间分成两个子空间[5]. M-tree 也是基于层次聚类划分空间，划分后数据空间里的数据聚集较稠密，这种结构非常适合检索的需求，数据空间划分示意图如图 3 所示.

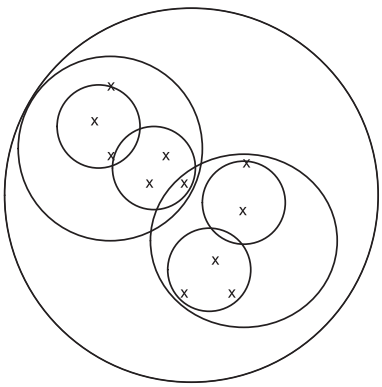


图 3　M-tree 数据分割

在安全过滤中，由于应用领域不一样，对数据处理的方式也不一样，基于检索的索引结构根本不能满足实时性需求. 下面我们将分析安全过滤中的数据分布特性，在此基础上描述具体过滤应用中需要解决什么样的特定问题才能构建适合过滤的高效索引结构. 假设对于给定数据集 X，其中包括有正例和反例，在实际网络环境中数据分布表现为一个非平衡分布特性，正例较多（正常信息），反例较少（敏感信息）. 在查询阶段，我们需要对反例进行判定，继而进行过滤处理. 在这样一个非平衡的数据分布条件下，利用否定性判断原理，只需快速否定判定大量正例数据，就可以加快整个查询的综合判定速度，从而提高索引结构过滤处理的性能. 下面通过一个简单实例来说明需要解决的问题. 考虑如图 4 所示的集合 S，包含有 9 个数据点，这些数据点都为反例数据，所需解决的主要问题是如何对数据区域进行划分使得集合 S 中的空白区域最大，即否定判定区域达到最大.

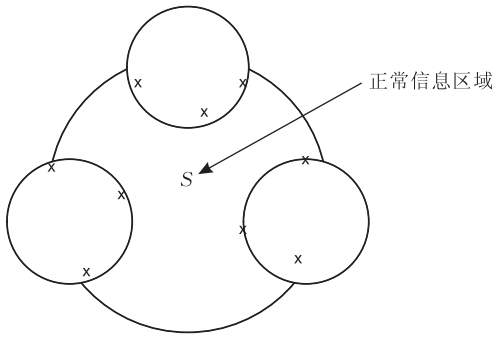


图 4　基本问题描述

通过上面区域划分构建索引结构，当给定查询点 q 时，查询数据点以最大概率落入划分区域后的空白区域，就可以快速判定查询数据点 q 必然不属于敏感信息. 由于在实际过滤中大部分数据都不属于敏感信息，所以这样的数据划分就使得大部分数据只需与根节点比较或与部分中间节点比较就可以得到最后判定结果，从而提高整体数据的查询性能.

想要快速得到判定结果，在区域空间划分时，就必须使得划分后的空白区域最大. 因此，在构建过滤索引树阶段，需要解决以下两个基本问题：

（1）如何得到包含所有点集的最小区域？

（2）如何确保每次区域划分后所得空白区域都最大？

在第 4 节，我们将具体描述如何通过最小圆覆盖的方法来解决基于否定性判断的快速过滤问题.

4　基于最小圆覆盖的过滤索引方法

4.1　最小圆覆盖算法

在数据划分时，为保证在每次区域划分中都得到最大空白区域，就需要用一个最小的区域来包括数据集簇，使得簇间较稀疏，空白区域达到最大. 求一个最小圆包含给定点集所有点的问题是人们在理论和实践上都十分感兴趣的一个问题[6-8]. 最小圆覆盖问题可以形式化描述为：给定有

限点集 S,包含所有点集 S 的最小圆 B 满足:$B=B(c,r):=\{x:\|x-c\|\leq r\}$. 由于最小圆的圆心是到集合中最远点的距离最近的一个点,因此在一些规划中有实际的应用价值. 圆心可以看作是点集的中心. 基于 n 个点的最小圆覆盖算法[9]可以用算法 1 描述.

算法 1. 最小圆覆盖算法.

输入:n 个数据点

输出:包含 n 个数据点的最小圆

1. 在点集 n 中任取 3 点 A,B,C.

2. 作一个包含三点的最小圆 C_1.

3. 在点集中找出距离 C_1 最远的点 D. 若点 D 在圆 C_1 的圆内或圆周上,则该圆即为所求的最小圆,算法结束. 否则,继续执行步 4.

4. 在 A,B,C,D 中选 3 个点,求解使包含这 4 个点的圆最小. 所选取的三点成为新的 A,B,C 三点,返回执行步 2.

最小圆覆盖算法在规划设施中有广泛应用. 在我们设计的过滤索引结构中,可以用来生成最大空白区域,从而可以加快搜索速度. 在 4.2 节我们将具体描述如何使用最小圆覆盖算法构建适合过滤的索引结构 F-tree.

4.2 F-tree 索引生成算法

构建适合 F-tree 过滤索引结构的主要目的是加快在数据查询阶段的判定速度,使得查询数据以最大概率落入索引结构所拥有数据点以外的区域,也就是上面所描述的数据划分中的空白区域. 因此在进行支点选择或区域划分时,首要考虑因素是相近点集之间组成的区域尽可能小,而划分后的点集区域之间的距离要尽可能大,才能保证在数据集划分后,整个集合区域所拥有的空白区域最大.

在实际安全过滤应用中,由于大量查询数据是正例,落入空白区域概率较大,因此设计具有较大空白区域的索引结构可以使大量查询数据只需查询树的根节点或部分中间节点就可以快速得到判定结果,而不需要遍历直到叶子节点才返回结果,从而加快查询数据集的整体查询速度. 由以上结合具体过滤应用领域的理论分析,基于启发式规则,我们设计了一种有效的基于最小圆覆盖的过滤索引结构 F-tree.

假设有包含 $n=3^k$ 的数据集 Ω,那么适合安全过滤应用领域的 F-tree 索引结构的构建方法由算法 2 给出. 在构建过滤索引树结构中,采用自低向上逐层构成节点的生成方式.

算法 2. F-tree_Build(Ω).

输入:反例数据集(Ω)

输出:F-tree

1. /* 叶子节点的生成 */

a) 在 Ω 中选择离原点最近的一点 u_1;

b) 寻找离 u_1 最近的两点 u_2、u_3;

c) 求解 u_1、u_2、u_3 最小覆盖圆 c_1;

d) 寻找离 c_1 最远的点 u_4;

e) 寻找离 u_4 最近的两点 u_5、u_6;

f) 求解 u_4、u_5、u_6 最小覆盖圆 c_2;

g) 重复执行步 d)~f),直到所有数据点都生成最小圆.

在通过以上处理阶段的区域划分后,可生成 3^{k-1} 个包含 3 个数据点集的最小覆盖圆,由这些最小圆集合可生成过滤索引树 F-tree 的叶子节点.

2. /* 中间层节点的生成 */

a) 从 3^{k-1} 个最小圆中选择圆心离原点最近的圆 c_1;

b) 寻找离 c_1 最近的两个圆 c_2、c_3;

c) 基于 c_1、c_2、c_3 包含的所有点,求解最小覆盖圆 c_4;

d) 寻找离 c_4 最远的两个圆 c_5;

e) 寻找离 c_5 最近的两个圆 c_6、c_7;

f) 求解 c_5、c_6、c_7 包含所有点的最小圆 c_8;

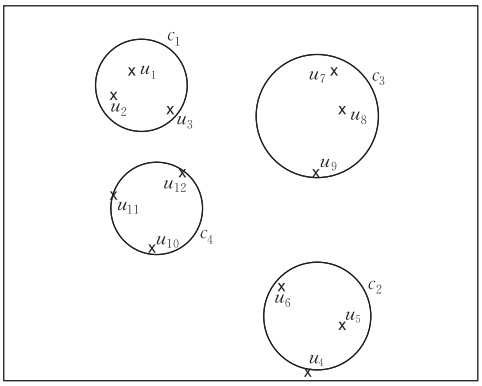
g) 重复执行步 d)~f),直到 3^{k-1} 个最小圆集合中每三个最小圆所包含的所有数据点都生成最小覆盖圆,可以生成 3^{k-2} 个最小覆盖圆.

通过叶子节点中 3^{k-1} 个最小圆之间结合处理,可以生成 3^{k-2} 个最小圆,这些最小圆覆盖了叶子节点中多个最小圆所包含的所有数据点集. 利用这个阶段生成的 3^{k-2} 个最小圆构成过滤索引树 F-tree 的第 $k-1$ 层中间节点.

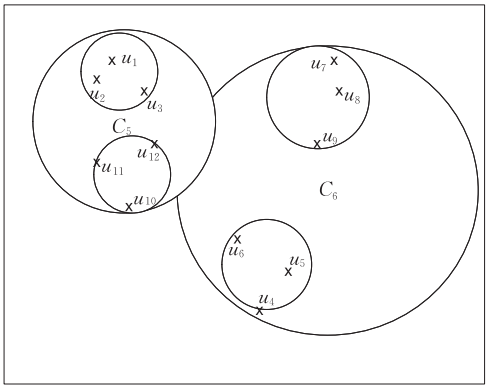
3. 重复第 2 阶段,经过迭代划分,依次可生成过滤索引树 F-tree 结构的第 $k-2,k-3,\cdots,1$ 层的中间节点.

4. 最后包含所有数据点的最小覆盖圆构成 F-tree 的根节点.

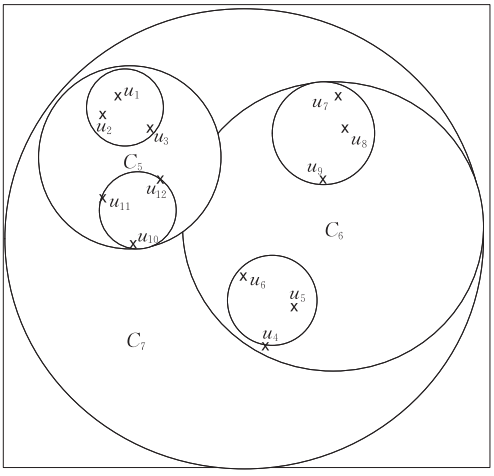
由上面建树过程可知,F-tree 为一颗平衡二叉树,具有平衡二叉树所拥有的良好特性,同时也具备具体应用领域的特点. 结合过滤应用领域中非平衡数据分布特性,利用最小圆覆盖划分数据区域,使得区域划分后的数据空间空白区域最大,符合查询阶段否定性判断的处理,可以加快整体的查询速度. 在图 5 中,通过一个简单的例子描述了过滤索引树 F-tree 构建过程中数据区域划分情况,首先在图 5(a)中,由 3 个数据点生成的最小覆盖圆构成 F-tree 中每个叶子节点;其次,F-tree 中间节点可以通过叶子节点中最小圆之间结合生成(图 5(b));最后覆盖所有点的最小圆构成 F-tree 的根节点(图 5(c)). 图示中只展示了两个最小圆的结合方式.



(a) 叶子节点生成示意图



(b) 中间节点生成示意图



(c) 根节点生成示意图

图 5　F-tree 区域划分示意图

4.3　F-tree 相似性查询算法

众多应用领域要求索引结构有相似性查询，即查找数据库中与某个给定的高维适量最近的 k 个数据，这一查询通常称为 K 近邻查询. 当 $K=1$ 时为“最近邻查询”，这个是在检索领域所需要的查询. 而在过滤应用领域，我们需要通过相似性来判定数据的特性或性质，不需要找出在查询阶段所搜索到的具体数据点. 因此，在 F-tree 的相似性查询阶段，采用区域查询的方法进行数据的判定处理. 区域查询可由定义 1 给出.

定义 1.　区域查询. 给定一个查询(q,r)，$q\in M$，M 为对象集，r 为查询半径，是一个非负值，区域查询就是要从对象集 M 中找出与查询 q 之间距离小于 r 的所有数据库对象.

给出区域查询的定义后，现在来描述基于区域查询的 F-tree 的查询算法. 给定查询 $R(q,r)$，设置查询 q 的查询半径为 r. 基于过滤索引 F-tree 的区域查询算法可用如下步骤来描述：

给定一个查询 $R(q,r)$

1. 从根节点开始遍历查询 F-tree.

2. 在每个中间节点 C_j，执行：

a) 如果 $d(q,C_j)\geq r$，退出搜索，对查询 q 执行放行处理.

b) 如果 $d(q,C_j)<r$，进入子节点搜索. 直到 q 不属于任意节点区域，退出搜索；否则，搜索到叶子结点，报告与 q 距离最近的节点 x_i.

与基于检索的索引结构不同，基于过滤的索引结构在查询阶段不需要遍历整棵索引树的所有节点，由于生成了最大的否定判定区域，大量查询在中间节点就可以报告满足条件的节点，退出查询. 在查询过程中，查询节点将大量减少.

5　实验结果与分析

5.1　实验设置

在实验中，我们对基于最小圆覆盖区域划分的索引结构 F-tree 进行了测试，并与基于检索的索引结构 M-tree 的性能进行了比较和分析. 两种索引结构均使用 Windows 环境下的 C 语言实现. 由于距离计算的费用较高，我们就使用“距离计算次数”作为算法计算复杂度的度量准则. 另一方面，由于我们实现的是内存索引结构，所以这里我们并不考虑磁盘的 I/O 操作. 在实验中，选取两个实验数据集，一个数据集 data1 包含 729 个随机生成的均匀分布数据，另一个是由 2187 个随机生成的均匀分布数据组成的数据集 data2. 分别验证所提出的过滤索引结构应用在不同数据量情况下的查询性能.

5.2　实验结果

数据集 1. 在数据集 1 的基础上，分别建立相应的 M-tree 和 F-tree. 查询性能的好坏是衡量索引结构好坏的一个重要标志[10]，在查询阶段，采用 $q(r)$ 区域查询方式. 我们随机地选取 100 个查询数据点，计算这 100 个查询的平均距离计算次数，并据此来比

较这两种索引结构的性能.在表 1 中,M-tree(times)和 F-tree(times)分别表示两种索引结构从根节点开始直到查询到满足要求的所有数据点所需要的距离比较次数.

表 1 两种索引距离比较次数(data1)

搜索半径 r	M-tree(times)	F-tree(times)
0.0001	16.51	27.88
0.002	18.77	28.24
0.004	21.18	28.78
0.006	23.57	29.56
0.008	25.97	30.28
0.01	28.08	30.70
0.02	39.67	33.61
0.04	61.80	42.28
0.06	82.36	53.26
0.08	101.77	65.86
0.1	121.10	82.90

将查询半径 r 的值设置为 0.0001 开始进行实验,在多组实验中,逐渐增加搜索半径 r 的值,直到 $r=0.1$,以此用来测试在设置不同搜索半径 r 的条件下,两种索引结构所需距离比较次数的变化.图 6 给出了数据集 data1 上两种索引所需要的距离比较次数随查询半径增加而产生的不同结果.

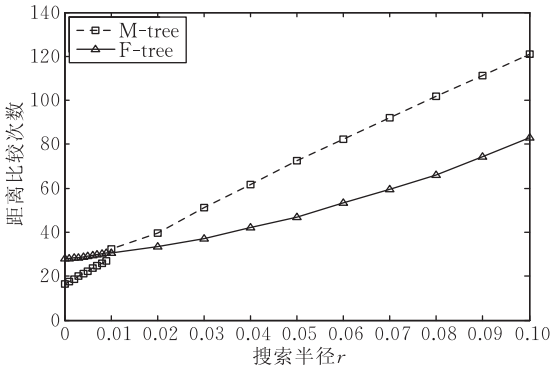


图 6 数据集 data1 上两种索引性能比较

从图 6 所展示的实验结果可以得出,在最初区域搜索半径为 0.0001 时,M-tree 索引结构的平均距离比较次数只有 16.51 次,而 F-tree 的平均距离比较次数却达到 27.88 次,F-tree 距离比较次数是 M-tree 的 1.6 倍左右.随着查询半径 r 的增加,F-tree 与 M-tree 之间的距离比较次数差距逐渐减小.当半径增加到 0.01 左右时,F-tree 的距离比较次数开始小于 M-tree,查询性能开始增强,继续增加查询半径,F-tree 与 M-tree 之间的距离比较次数差距变大,在整个查询半径变化过程中,F-tree 的查询性能整体呈现增强趋势,查询性能优于 M-tree.总体来说,r 在 0～0.1 的区间变化时,M-tree 的查询性能受半径变化的影响较大,表现为图 6 中斜率较大的曲线.相反,F-tree 在整个半径变化过程中,查询中所需距离比较次数随着半径增加,所需的查询次数变化较小,在整个查询半径增加的过程中,性能趋于稳定.因此,在一定查询半径变化波动范围内,F-tree 在查询性能上相对于 M-tree 来说,性能相对稳定.

数据集 2. 主要测试两种索引结构处理大数据集的性能变化情况.同样,我们从数据集中随机选择 100 个数据作为测试中的查询数据,表 2 中给出了两种索引结构在不同查询半径设置下所需的距离比较次数.图 7 描述了两种索引结构所需距离比较次数随查询半径变化而变化的曲线图.

表 2 两种索引结构距离比较次数(data2)

搜索半径 r	M-tree(times)	F-tree(times)
0.0001	40.94	34.09
0.002	45.33	34.75
0.004	50.15	35.98
0.006	54.89	37.33
0.008	59.30	38.50
0.01	63.63	39.64
0.02	85.76	46.63
0.04	129.72	69.10
0.06	173.20	97.48
0.08	215.44	133.42
0.1	258.26	175.99

从表 2 和图 7 所展示的查询结果可以得出,当搜索半径为 0.0001 时,M-tree 索引结构所需的距离计算次数为 40.94,而 F-tree 却只需要 34.09,在最初始设定的查询半径条件下,F-tree 所需的距离比较次数要小于 M-tree.而随着查询半径的增加,F-tree 所需距离比较次数与 M-tree 所需的距离比较次数之差也逐渐增加,距离比较次数只有 M-tree 的 1/2 左右.在整个查询半径变化区间内,F-tree 距离比较次数随查询半径增加的变化相对稳定,在图 7 上表现为斜率较小的曲线.

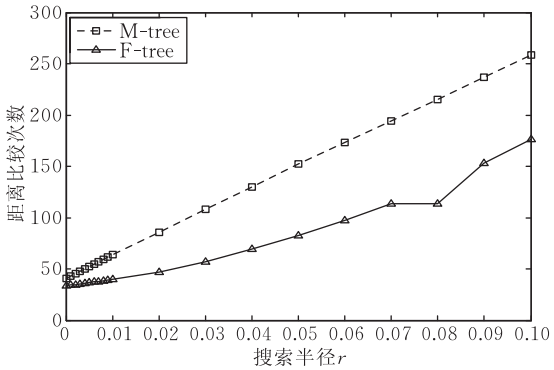


图 7 数据集 data2 上两种索引性能比较

下面，我们比较数据集大小的变化对查询性能的影响，设定搜索半径为 $r=0.02$，分别得到在两种数据集下所构建的 M-tree 和 F-tree 两种索引结构的查询性能比较（图 8）. 从图 8 中的结果描述中可以得出，在搜索半径设置为 0.02 的情况下，F-tree 的搜索性能在两种数据集上的测试中都要优于 M-tree. 尤其是应用在不同数据量的情况下，M-tree 搜索所需要的距离比较次数随着数据量急速增大，也表现急剧增加的趋势：当数据量为 729 时，所需距离比较次数为 39.67，而当数据量增加到 2187 时，所需的距离比较次数增加到 85.76，这个距离比较次数是应用于小数据集时所需距离比较次数的 2 倍左右，查询性能下降速度较快. 而对于所提出的基于过滤的索引结构 F-tree 来说，从图 8 所展示的实验结果可以得出，当数据量为 729 时，距离比较次数为 33.61，而当数据量增加到 2187 时，所需的距离比较次数为 46.63，数据量的增加对 F-tree 的索引性能影响很小.

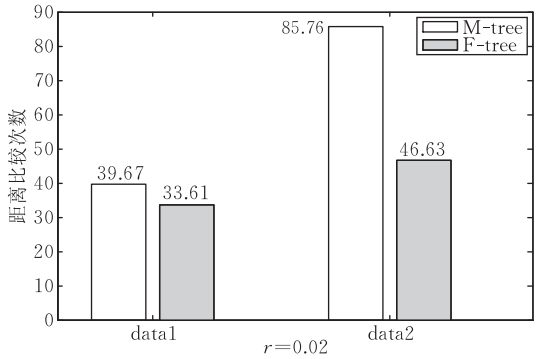


图 8　两种数据集 data1 和 data2 的性能比较

5.3　结果分析

在所设置的多组实验中，F-tree 都表现出了较好的性能，这是因为我们所需要面对的具体问题是敏感信息过滤，而实际网络环境中敏感信息数据分布具有非平衡性，并且在过滤过程中需要具有较快的判定速度. 根据以上两个特性，利用最小圆覆盖划分数据区域，使否定区域达到最大，这样构建的索引树是一棵平衡树，每个节点区域不会存在覆盖问题，所划分的节点区域也会比较稀疏. 在 F-tree 的查询阶段，由于我们得到了最大的否定区域，在查询半径较小的情况下，需要比较的数据点较多，查询性能优势不明显. 而当半径增加，查询 q 遍历节点时，覆盖的数据点较多，分布较多的正常数据以最大概率落入否定区域，从而在查询时可以快速得到判定结果. 而基于检索的 M-tree 索引结构在划分数据空间时，数据划分比较稠密，以便搜索到更多的相似性数据，在查询判定阶段，每到一个区域都需要进一步判断，并不适合需要快速判定的非平衡分布敏感信息过滤领域，在搜索距离比较次数上相比稀疏区域划分的过滤索引结构 F-tree 有所增加.

6　结束语

索引是基于内容相似性查询的有效方法，而数据空间的分割直接影响索引结构的查询性能. 本文针对实际敏感信息过滤应用中数据非平衡分布的特点，提出了基于否定判断的过滤模型. 采用基于最小圆覆盖的区域划分方法解决基于否定性判断的过滤问题，构建符合否定判断的过滤索引结构 F-tree. 实验结果显示：在不同查询半径值 r 设置下，F-tree 的查询性能明显优于 M-tree，并且随着数据量的增加，F-tree 所需的距离比较次数增长幅度不大. 因此，所设计的 F-tree 索引结构在综合性能上表现出一定的稳定性和鲁棒性.

参　考　文　献

[1] Bustos B, Navarro G, Chávez E. Pivot selection techniques for proximity searching in metric spaces. Pattern Recognition Letters, 2003, 24(14): 2357-2366

[2] Bustos B, Pedreira O, Brisaboa N. A dynamic pivot selection technique for similarity search//Proceedings of the 1th Workshop on Similarity Search and Applications. Washington, USA, 2008: 105-112

[3] Chávez E, Navarro G. A compact space decomposition for effective metric indexing. Pattern Recognition Letters, 2005, 26(9): 1363-1376

[4] Ciaccia P, Patella M, Zezula P. M-tree: An efficient access method for similarity search in metric spaces//Proceedings of the International Conference on Very Large Data Bases. San Francisco, USA, 1997: 426-435

[5] Zhou Xiang-Min, Wang Guo-Ren. Key dimension based high-dimensional data partition strategy. Journal of Software, 2004, 15(9): 1361-1374(in Chinese)
(周项敏，王国仁. 基于关键维的高维空间策略. 软件学报，2004，15(9)：1361-1374)

[6] Fischer K, Gartner B, Kutz M. Fast smallest-enclosing-ball computation in high dimensions//Proceedings of the 11th Annual European Symposium on Algorithms. Budapest, Hungary, 2003: 630-641

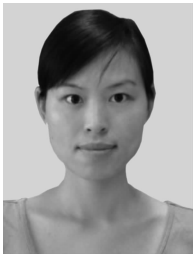
[7] Xu S, Freund R, Sun J. Solution methodologies for the smallest enclosing circle problem. Computational Optimization and Applications, 2001, 25(1): 283-292

[8] Fan Ke-Lei. Research on approximate minimum enclosing

balls in high dimensions [M. S. dissertation]. Shandong University, Shandong, 2010(in Chinese)
(范克磊. 高维空间近似最小球覆盖问题的研究[硕士学位论文]. 山东大学, 山东, 2010)

[9] Wang Wei, Wang Wen-Ping, Wang Jia-Ye. An algorithm for finding the smallest circle containing all points in a given point set. Journal of Software, 2000, 11(9): 1237-1240(in Chinese)
(王卫, 王文平, 汪嘉业. 求一个包含点集所有点的最小圆的算法. 软件学报, 2000, 11(9): 1237-1240)

[10] Chávez E, Navarro G, Baeza-Yates R et al. Searching in metric spaces. ACM Compute Surveys, 2010, 33(3): 273-21



CHEN Jie, born in 1982, Ph. D. candidate. Her main research interests include information security, similarity search, and index technology.

FANG Bin-Xing, born in 1960, Ph. D., professor, Ph. D. supervisor. His current research interests include information security.

TAN Jian-Long, born in 1974, Ph. D., professor, Ph. D. supervisor. His current research interests include information security, similarity search, and index technology.

JIN Shi-Chao, born in 1988, M. S. candidate. His current research interests include index schemes, similarity search.

Background

This paper focuses on the research of filtering algorithm in information content security. Filtering algorithm design is an important step for information filtering system. Some methods have been proposed to apply pattern classification, such as SVM, Bayes, et al, to filter unmoral data. However, these methods do not work well to be faced with massive data. Then, some researches proposed index-based method to cope with these massive data, but at present, these indexes only consider the information retrieval application, some data area partition methods do not suitable for the good performance demand of filtering. In this paper, an index filtering algorithm based on minimum enclosing circle partition is presented. Considering the imbalance data distribution in real filtering situation which including most normal data and rare unmoral data. In this situation, we need to filter unmoral data with a good performance. By utilizing minimum enclosing circle for unmoral data area partition, we can get the biggest negative area, so a filtering index structure F-tree will be built. In the query step, all query data will be taken into negative is with the maximum probability. According to the principle of negative judgment, the query data need not to search all nodes in F-tree, and this searching may be exit at some intermediate node. Therefore, our method can achieved a good filtering performance owing to the fast judgment for the most normal data. This work is partly supported by the National High Technology Research and Development Programs (863 Program) of China under Grant No. 2011AA010705 and No. 2007CB31110. This algorithm is meaningful for improving the performance of filtering system. The team has done several works in the field of information content security. A filtering prototype system has built, and the feature extraction work has been discussed in some other papers. This paper focuses on the filtering algorithm for improving the filtering speed.