基于差分隐私保护的 KDCK-medoids 动态聚类算法

马银方 张 琳
(南京邮电大学计算机学院 南京 210003)

摘 要 K-medoids 算法对初始中心点敏感,不能有效地对动态数据进行聚类,且需要对相关的隐私数据进行保护。针对这些问题,提出了基于差分隐私保护的 KDCK-medoids 动态聚类算法。该算法在采用差分隐私保护技术的基础上将 KD-树优化选取出的 k 个聚类中心和增量数据相结合建立新的 KD-树,然后采用近邻搜索策略将增量数据分配到与其相应的聚类簇中,从而完成最终的动态聚类。通过实验分别对小数据集和多维的大数据集的聚类准确率及运行时间进行了分析,同时也对采用差分隐私保护技术的 KDCK-medoids 算法在不同数据集上的有效性进行了评估。实验结果表明,基于差分隐私保护的 KDCK-medoids 动态聚类算法能够在实现隐私保护的同时快速高效地处理增量数据的动态聚类问题。

关键词 KD-树,K-medoids 聚类算法,差分隐私,动态聚类
中图法分类号 TP393 **文献标识码** A

KDCK-medoids Dynamic Clustering Algorithm Based on Differential Privacy

MA Yin-fang ZHANG Lin
(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract The traditional K-medoids clustering algorithm is sensitive to the initial center points, can't effectively deal with dynamic data clustering, and needs privacy protection for private data. Therefore, this paper proposed the KDCK-medoids dynamic clustering algorithm. It establishes a new KD-tree using kth rectangular units optimally selected by KD-tree and incremental data based on differential privacy protection technologies, and then distributes the incremental data into the corresponding clusters by using the neighbor search strategy, and then completes the dynamic clustering. Through experiments on small data sets and multi-dimensional large data sets, clustering accuracy and running time are analyzed. And the effectiveness of the algorithm is evaluated. The experimental results indicate that the KDCK-medoids dynamic clustering based on differential privacy protection can realize privacy protection meanwhile quickly and efficiently process the dynamic clustering of incremental data problem.

Keywords KD-tree, K-medoids clustering algorithm, Differential privacy, Dynamic clustering

1 引言

伴随着互联网以及信息技术的飞速发展,大量的数据在各个领域不断地累积。随着时间的推移,很多领域都产生了庞大的数据量。对海量数据进行挖掘并对动态的数据进行聚类能够获取大量非常有价值的信息,与此同时也要注意对相关的隐私信息进行有效的保护。因此,对海量数据进行动态聚类,同时引入隐私保护技术,具有重要的研究意义,也将有很大的应用前景。

K-medoids 聚类算法是基于划分的算法,是比较经典且应用较为广泛的算法。传统 K-medoids 算法的初始中心点是在数据集中随机选取的,因而对初始中心点较为敏感。众多学者针对 K-medoids 聚类算法对初始中心点敏感这一问题进行了大量的实验研究,并对其改进后的算法进行了不断的探索。文献[1]提出了对初始中心点进行微调,并提出了增量中心候选集这一概念,从而对 K-medoids 算法作出了改进。文献[2,3]提出了一种新的基于差分演化的 K-medoids 聚类算法,该算法缩短了收敛的时间,同时改善了聚类质量,有效地克服了 K-medoids 聚类算法的缺点。文献[4,5]也提出了在大数据集群中搜索最优中心点的改进算法。随着海量信息时代的到来,每秒钟都能产生非常庞大的数据量,而且数据复杂多变,因此对不断变化的动态数据进行聚类具有重要的探索价值。文献[6]提出了利用聚类集成对数据进行分类的增量聚类算法,其能够有效地对增量数据进行聚类。文献[7]中提出了采用增量和层次两个方面相结合的高效聚类方法。文献[8-10]中对增量聚类算法进行了重大的改进,提出了快速稳定的增量聚类算法。

传统的 K-medoids 聚类算法应用较为广泛,但不能高效地处理动态聚类问题,大量的隐私数据容易受到外部的各种攻击,因此引入隐私保护技术具有非常重要的作用。针对这

本文受国家自然科学基金(61402241,61572260,61373017,61572261,61472192),江苏省科技支撑计划(BE2015702)资助。
马银方(1989-),女,硕士生,主要研究方向为信息安全、数据挖掘、隐私保护等;张 琳(1980-),女,博士后,副教授,硕士生导师,主要研究方向为云计算、网络安全、信任、可信计算等,E-mail:zhangl@njupt.edu.cn。

些问题，本文提出了基于差分隐私保护的 KDCK-medoids 动态聚类算法，该算法在采用差分隐私保护技术的基础上将 KD-树优化选取出的 k 个聚类中心和增量数据相结合建立新的 KD-树，然后采用近邻搜索策略将增量数据分配到与其相应的聚类簇中，从而完成最终的动态聚类。该算法能够在实现隐私保护的同时快速高效地处理增量数据的动态聚类问题。

2 KD-树优化中心点选取方法

KD-树(即 k-dimensional tree)，它表示一种数据结构，能够将数据节点划分到 K 维空间中。KD-树实质上是二叉树，而它的每个节点表示的是一个空间范围。由于初始中心点是随机选取的，并不能反映数据真实的分布情况，因此采用传统的 K-medoids 聚类方法得到的聚类结果往往是不稳定的。由于是随机选取，不可避免会选到孤立点作为初始聚类中心，这不利于数据进行聚类，从而影响聚类的效率和最终聚类结果，因此选取的初始中心点较为分散比较好。如果首先采用 KD-树存储结构对数据进行预处理，则可以有效地改善初始中心点选取的效果。也就是说，采用 KD-树对待聚类数据进行优先处理，然后根据 KD-树空间划分思想有方向性地选取出初始中心点。由于 KD-树划分的空间区域在一定程度上可以反映整个数据集的实际数据分布情况，因此可以得知采用 KD-树优化中心点选取方法比随机选取初始聚类中心点更有效，且准确率较高。因此本文提出了 KD-树优化中心点选取方法。

为了进一步研究 KD-树优化中心点选取方法，定义以下几个公式，首先设样本数据集

$$A=\{a_1,a_2,\cdots,a_n\}$$

定义 1 单个矩形单元中所包含的数据元素的个数 Num 为

$$Num=\frac{n}{m\times k} \tag{1}$$

其中，n 表示样本数据集元素的个数，k 为聚类的个数，m 表示子块个数，即一个聚类被分割成的子块的数目，该数据可根据所给数据集的大小作出适时的调整，通常在数据集样本数目差别不大的情况下，可以把 m 取为 10。通过 n,m,k 能够构造出完整的 KD-树，而 k,m 两个参数则能够反映出 KD-树的深度和所包含的叶子节点的数目。

定义 2 矩形单元中心 C_i 为

$$C_i=\frac{S_i}{W_i} \tag{2}$$

其中，S_i 表示该矩形单元中所有元素的线性和；W_i 表示该矩形单元的权重，主要用该矩形单元中所包含的样本元素的个数来表示。

定义 3 矩形单元的密度 Den，主要用来表示矩形单元中所包含的数据元素之间的密集程度。

$$Den_i=\frac{W_i}{V_i}=\frac{W_i}{(\max(d_{\max}-d_{\min}))^2} \tag{3}$$

其中，W_i 表示矩形单元中包含的样本元素的数目；V_i 表示矩形单元的面积；$d_{\max}$，$d_{\min}$ 表示相应的矩形单元中数据元素的最大值及最小值。

KD-树优化选取初始中心点的算法的具体描述如下。

输入：样本数据集

输出：k 个优化聚类中心点

1. 根据样本数据集 A 的大小和聚类数 k 来确定分割子块 m 的数目，从而能够确定 KD-树的深度，进而构建出 KD-树。
2. 通过 KD-树的结构来分割数据集中的数据元素，计算每个矩形单元样本元素的个数，同时也需计算矩形单元的中心以及矩形单元密度。
3. 根据所得出的矩形单元密度进行降序排列，从而形成新的数据集 A*。
4. 最终选取出数据集中的前 k 个数据对象作为优化聚类中心点。

3 基于差分隐私的 KDCK-medoids 及其动态聚类方法

3.1 差分隐私保护技术在聚类算法中的应用

由文献[11,12]可知，差分隐私保护技术是基于数据随机加扰的隐私保护方法，主要是通过添加噪声来对敏感数据进行隐藏的一种隐私保护方法。在差分隐私保护方法中，具体某个记录的变化对整个数据集的计算结果没有太大的影响，也就是说，单个数据记录在该数据集中或者不在该数据集中对数据集最终的计算结果的影响非常小。因此，攻击者无法通过观察计算结果而获取准确的个体信息，而且差分隐私保护模型是一种严格的隐私保护模型，它不在乎攻击者拥有多少相关的背景知识，哪怕攻击者拥有了某一条记录之外的其他所有的数据信息，也不会因此而使另外一条记录的信息泄露。所以，差分隐私保护技术可以有效地对一些敏感的隐私信息进行保护，且具有很好的可用性。

设有包含 n 个数据对象的样本数据集 $A=\{a_1,a_2,\cdots,a_n\}$，首先采用 KD-树优化中心点选取算法选取出 k 个优化中心点构成集合 $C=\{c_1,c_2,c_3,\cdots,c_k\}$，然后对这 k 个优化中心点进行噪声加扰，所加的 Laplace 噪声函数为 $L(b)=e^{(-|x|/b)}$，其中 $b=\Delta f/\varepsilon$，返回加扰后的 k 个点为 $\{c_1',c_2',c_3',\cdots,c_k'\}$，选取这 k 个点作为新的初始中心点，采用以下方法进行迭代更新：

(1)将样本数据集 A 中其它的对象分别分到与其距离最近的初始中心点所代表的簇中，从而形成 k 个相对独立的簇。

(2)在 $1\leqslant i\leqslant k$ 内随机选择一个有中心点 c_i' 的簇，在此簇内随机选择一个非代表对象 c_{random}，并加上噪声 $L(b)=e^{(-|x|/b)}$，其中 $b=\Delta f/\varepsilon$，返回一个 c'_{random}，然后用 c'_{random} 替换原簇内的代表对象 c_i'，计算绝对误差之和 s'，并用 s' 减去 s 得到一个结果 Q，其中

$$S=\sum_{i=1}^{k}\sum_{p\in A_i}d(p,c_i') \tag{4}$$

其中，A_i 表示第 $i(i=1,2,\cdots,k)$ 个初始聚类所形成的数据集，p 为 A_i 中的任一数据对象，c_i' 为 A_i 中的代表对象即中心点，S 表示该数据集中的每一个对象 p 与 A_i 中的代表对象 c_i' 的绝对误差之和。

(3)如果结果 $Q<0$，则这次替换可行，用 c'_{random} 去替换 c_i'，重新分配形成新的 k 个代表对象的集合；如果结果 $Q>0$，则 c_i' 作为这个簇的代表对象是可以被接受的，此时不发生任何变化。

按照以上 3 个步骤进行循环迭代，直至最终 k 个簇内的每个对象都不再发生任何变化为止。

3.2 动态聚类算法

随着互联网的快速发展，新数据无时无刻不在增长着，且数据也在不断地变化着。由于传统的聚类算法仅适用于静态数据，但对于动态的数据以及一些增量数据而言，随着数据的更新，聚类的结果往往也不稳定，而且需要对所有的数据重新进行聚类，这就降低了聚类的效率，同时也浪费了大量的计算

资源。围绕中心点划分(Partitioning Around Medoids,PAM)聚类算法,其每次迭代的时间复杂度为 $O(k(n-k)^2)$(其中 n 是数据对象的数目,k 是聚类数),因此 k-中心点或 PAM 算法对于较小的数据集非常有效,但不能很好地扩展到大型数据集中;而且当数据集中的数据对象由于有一定的更新而发生相关的变化时,最终聚类的结果也会发生相应的变化。对于大型数据集而言,在更新后的新的数据集上再重新执行一遍聚类算法以相应地更新聚类结果显然是比较低效的,因此有效地采用增量式聚类算法显得非常重要。文献[13]提出了一种改进的增量式 K-medoids 聚类算法,该算法能够很好地解决动态数据更新时所面临的一些问题。本文提出了 KD-树优化的 K-medoids 聚类算法,其主要是采用 KD-树的数据结构来存储数据集中所有的数据对象,并通过 KD-树优化选取出 k 个稳定的初始中心点,然后采用近邻搜索策略可以很快地寻找到给定的数据对象的近邻对象,可以减少搜索时间,并且能够快速高效地对动态数据进行聚类。

文献[14]中提出了基于密度的增量聚类算法,该算法可以处理动态数据,同时也提高了数据资源的利用率。文献[15]中提出的 KDTK-means 聚类算法也能够有效地处理动态数据的聚类问题,对传统的聚类算法进行了很大的改进。本文中提出了 KDCK-medoids 动态聚类算法以更好地对动态数据以及增量数据进行聚类。为了研究 KDCK-medoids 动态聚类方法,定义以下两个距离。

定义 4(簇间距离) 设有两个样本数据集 A_i,A_j,用 $Dis(A_i,A_j)$来表示任意两个样本簇之间的距离,定义如下:

$$Dis(A_i,A_j)=\frac{1}{m_1*m_2}\sum_{i=1}^{m_1}\sum_{j=1}^{m_2}Dis(x_i,x_j) \tag{5}$$

其中,x_i,x_j 为 A_i,A_j 中的任意元素,m_1,m_2 分别表示样本数据集 A_i,A_j 中包含元素的个数,$Dis(A_i,A_j)$是用欧氏距离来表示两个样本数据 x_i,x_j 之间的距离。

定义 5(平均簇间距离) 采用聚类算法所产生的簇与簇之间的平均距离,定义如下:

$$Mdis(A)=\sum_{i=1}^{k}\sum_{j=1}^{k}\frac{Dis(A_i,A_j)}{C_k^2} \tag{6}$$

其中,$Mdis(A)$表示样本数据集 A 采用聚类算法之后所生成的各个簇之间的平均距离;A_i,A_j 表示所生成的任意两个簇;k 表示采用聚类算法之后所产生的聚类的个数。

KDCK-medoids 动态算法的具体描述如下。

输入:包含 n 个数据元素的原始样本数据集 $A=\{a_1,a_2,\cdots,a_n\}$和包含 m 个元素的增量数据集 $C=\{c_1,c_2,\cdots,c_m\}$

输出:最优的聚类结果

1. 采用 KD-树优化算法选取出 k 个初始中心点,设为 $S=\{s_1,s_2,\cdots,s_k\}$,当有新的增量数据 $C=\{c_1,c_2,\cdots,c_m\}$出现时,用这两个数据集中的元素来共同构建一个新的 KD-树;
2. 假设第 i 个聚类中所包含的增量数据的个数为 Q_i,同时假设该聚类中增量数据出现之前的原数据对象所组成的集合记为 G_i;
3. 对于各个增量数据 $C_i(i=1,2,\cdots,m)$,搜索其对应的最近邻的初始中心点 $S_i(i=1,2,\cdots,k)$,然后将增量数据分配到与之相应的集合 G_i 中;
4. 递归搜索每个增量数据 $C_i(i=1,2,\cdots,m)$所对应的最近聚类中心 $S_i(i=1,2,\cdots,k)$,然后采用 K-medoids 算法逐步更新第 i 个聚类中心,从而形成 k 个新的聚类簇;
5. 比较 k 个聚类簇任意两个簇之间的簇间距离,如果两个簇的簇间距离小于平均簇间距离,则对两个簇进行合并,直至最终实现任意两个簇的簇间距离均大于平均簇间距离为止;
6. 输出最终的聚类结果。

该算法采用 KD-树优化选取出 k 个聚类中心,当有增量数据出现时,把 k 个初始中心和增量数据放在一起来构建新的 KD-树,然后采用近邻搜索策略为增量数据寻找相应的最近邻,从而将其分配到相应的簇中。增量分配到相应的簇中之后,再根据平均簇间距离合并相临近的簇,也就是说,如果有两个簇之间的簇间距离是小于平均簇间距离的,则需合并这两个簇;如果两个簇的簇间距离大于或等于数据集中的平均簇间距离,则不需要对其进行合并。之后依次比较其他的簇,直到任意的两个簇之间的簇间距离均大于平均簇间距离为止,这样能够进一步修正增量分配的结果,从而有效地使聚类结果达到最优。该算法只需要对增量数据进行相关的处理,从而避免了在出现增量数据时对全部的数据进行重新聚类,因而在一定程度上提高了对增量数据进行聚类的效率。

4 实验结果及分析

本实验分 3 部分对新提出的方法进行实验分析:1)对 KD 树优化选取中心点的算法进行实验分析;2)KDCK-medoids 动态聚类算法的相关实验分析;3)采用差分隐私保护技术的 KDCK-medoids 算法的有效性进行实验分析。为了更好地对实验结果进行分析,本实验是在假设每个聚类的细分因子 m 相同的情况下进行的。

对基于 KD 树优化选取初始中心点算法的效率和可用性进行分析,所有的实验结果均是在 Matlab 下模拟仿真所得。实验所使用的数据集来自 UCI Machine Learning Repository (archive.ics.uci.edu/ml/database.html),采用的数据集为 UCI 中的 Iris,Ecoli,Acute Inflammations,Breast cancer,Thyroid 数据集。数据集的具体描述细节如表 1 所列。

表 1 数据集的构成描述

数据集名称	数据类型	数据集记录数	属性数	聚类数
Iris	Multivariate	150	4	3
Ecoli	Multivariate	336	8	7
Acute Inflammations	Multivariate	120	6	3
Breast cancer	Multivariate	699	10	2
Thyroid	Multivariate	220	6	4

对 KD 树优化选取中心点算法与传统 K-medoids 聚类算法 1 在进行相同聚类时的准确率进行了相关的比较,比较结果如表 2 所列。

表 2 KD 树优化选取算法与传统 K-medoids 算法的准确率比较

数据集	K-medoids 算法		KD 树优化算法	
	运行时间(ms)	准确率(%)	运行时间(ms)	准确率(%)
Iris	35	75.32	46	86.51
Ecoli	78	72.42	107	84.53
Acute Inflammations	42	81.62	54	85.76
Breast cancer	85	92.38	98	95.52
Thyroid	72	78.65	85	83.42

从表 2 中可以看出,文中提出的 KD 树优化中心点选取算法与传统的 K-medoids 算法相比准确率得到了明显提高,这充分说明了采取 KD 树优化选取中心点算法是有效的。但在实验过程中,KD 树优化选取算法由于需要建立 KD 树及计算矩形单元中心和矩形单元密度等,因此时间消耗比较大,这是不可避免的;而传统 K-medoids 算法只是随机选取初始

中心点，所以其时间开销比较小。由此可见，本文中所提出的KD树优化选取算法与传统K-medoids相比，在低维数据的处理上有较高的准确率。

下面采用人工数据集D1－D6进一步对文中提出的KD树优化选取算法对高维数据的有效性进行相关的验证。数据集的具体描述如表3所列。

表3　人工数据集D1－D6

数据集	数据集记录数	数据维度	聚类数
D1	4601	5	12
D2	4601	10	12
D3	4601	15	12
D4	4601	20	12
D5	4601	25	12
D6	4601	30	12

将上述数据分别应用于本文提出的KD树优化选取算法与传统的K-medoids算法1，并对每组数据进行了5次独立的实验，同时对其准确率进行了详细的分析。对每组数据取5次实验结果的平均值并进行了相关的记录，实验分析所得结果如图1所示。

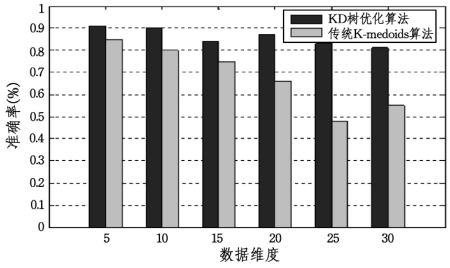

图1　不同维度下的准确率分析

从图1可以看出，对于低维数据而言，传统K-medoids算法与本文提出的KD树优化算法都有比较高的准确率，而当数据维度不断增大时，传统K-medoids算法的聚类准确率明显降低，而KD树优化算法仍能够保持较高的准确率。因此可得，文中提出的KD树优化算法也适用于高维的数据。当然，在数据维度越高的情况下，时间开销也会增加。

本文提出的KDCK-medoids动态聚类算法主要采用的是近邻搜索策略对增量数据进行相关的聚类，与当前常用于处理增量数据的增量DBSCAN算法[16]进行相关的比较。实验采用人工数据集D7，D7为含有1000个数据的10维数据集，对其进行相关的聚类，在增量数据的数目不断增加时，采用这两种方法对增量数据进行聚类时所需要的运行时间以及准确率进行了比较。对每组数据进行了5次独立实验并计算其平均值，得出的实验结果如表4和图2所示。

表4　运行时间和准确率的比较

增量数据数目	KDCK-medoids算法		增量DBSCAN算法	
(K)	运行时间(s)	准确率(%)	运行时间(s)	准确率(%)
1	0.32	95.27	1.58	90.28
2	0.48	92.31	1.96	87.68
3	0.75	86.62	2.53	83.43
4	0.87	85.92	3.25	86.53
5	0.97	85.25	3.46	77.96
6	1.25	82.43	3.58	74.56
7	1.56	81.35	4.52	80.28
8	1.72	80.82	4.85	88.16
9	1.92	80.23	5.36	80.73
10	2.36	79.83	6.42	76.54

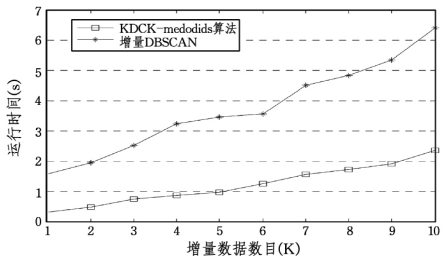

图2　KDCK-medoids算法与增量DBSCAN算法在增量数据数目不断增加时聚类所用的时间对比分析

通过表4和图2的分析可以得知，本文中提出的KDCK-medoids算法与增量DBSCAN相比有着较高的准确率且时间开销比较少，因此采用本文的KDCK-medoids动态聚类算法处理数据更新时效率更高。

为了对采用了差分隐私保护技术的KDCK-medoids聚类算法的有效性进行相关的分析，采用文献[16]中所提出的F-measure评价指标。F-measure主要用于评判改进后的算法的聚类结果与改进之前的差别。如果F-measure的结果为1，则表示添加了噪声之后的算法能够在保护隐私数据的同时使得聚类的最终结果没有任何改变，也就是说F-measure的值越接近1，改进的算法就越成功，有效性越高。为了便于对比，简单定义几个计算公式。

首先计算出数据集直接使用不添加噪声的原始数据进行KDCK-medoids聚类所得的结果，记为S；然后计算出采用差分隐私保护技术添加噪声后进行KDCK-medoids聚类所得的结果，记为S'。

设数据样本集共有N个对象，用A_i来表示S中的任意一个簇，用C_j来表示S'中的任意一个簇，定义一个n_{ij}来记录A_i和C_j中有多少相同的簇。再定义两个参数p_1和p_2，其中p_1表示n_{ij}占S全部簇数的比例，p_2表示n_{ij}占S'全部簇数的比例。具体公式如下：

$$F(S')=\sum_{A_i\in S}\frac{|A_i|}{N}\max\{F(A_i,C_j)\} \tag{7}$$

其中，$F(A_i,C_j)=(2*p_1*p_2)/(p_1+p_2)$。

差分隐私保护算法中作为隐私保护预算的ε的值直接决定了该算法的隐私保护程度，它的大小也表示了添加噪声的多少，ε的值越小表示添加的噪声越多，同时隐私保护的水平也就越高。所以可以随着ε的值的变化来观察F-measure的改变来衡量隐私保护的有效性，从而通过ε的改变来评价在不同噪声添加量的情况下各隐私保护算法的聚类可用性程度。

该实验采用的UCI中的数据集为：Wine，Haberman，Waveform Database，MAGIC。数据集的具体描述如表5所列。

表5　数据集的构成描述

数据集	数据类型	属性数	记录数	数据集代号
Wine	Multivariate	13	178	D1
Haberman	Multivariate	4	306	D2
Waveform Databas	Multivariate	40	5000	D3
MAGIC	Multivariate	11	19020	D4

由于对每个数据集添加的噪声具有随机性，对每个数据集采用隐私保护以及没有添加噪声之前的KDCK-medoids在D1－D4数据集上所得出的F-measure的值的情况进行了分析。为了防止出现较大的误差从而最大程度争取计算结果的

准确性，分别在每个数据集上进行了 10 次实验并取结果的平均值，对结果进行了对比分析，如图 3 和图 4 所示。

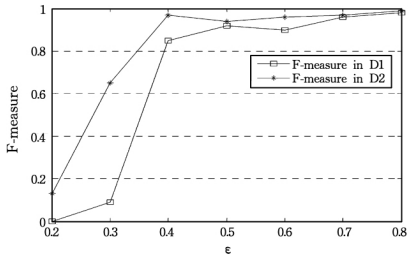

图 3　KDCK-medoids 在数据集 D1 和 D2 上的 F-measure 值

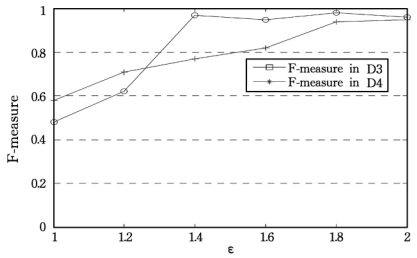

图 4　KDCK-medoids 在数据集 D3 和 D4 上的 F-measure 值

从图 3 和图 4 可知，差分隐私 KDCK-medoids 聚类算法满足 ε-差分隐私保护的要求，并且能够有效地保护个人隐私。从图中可以得知，差分隐私 KDCK-medoids 聚类算法仅需要添加少量的噪声就可以达到很好的隐私保护效果。综合分析图 3 和图 4 可知，在相同隐私保护级别下（即 ε 的值相同的情况下），小数据集的有效性高于大数据集的，低维数据集的有效性高于高维数据集的。隐私保护的级别 ε 的值是可以根据需求适时地调整的，可以通过对其值进行调整来实现不同程度的隐私保护。

以上 3 部分的实验结果表明，本文提出的基于差分隐私保护的 KDCK-medoids 动态聚类算法既能够对低维数据进行动态聚类，也能够对有增量数据的高维数据进行动态聚类，并且具有较高的准确率，同时引入了差分隐私保护技术有效地保护了隐私数据。该算法的思想对未来更好地处理海量数据中增量数据的动态聚类和对其中的隐私数据进行保护具有一定的应用价值。

结束语　传统的聚类算法大多都是对静态的数据进行聚类，而现实生活中的数据是不断变化和增长的，也就是说对动态的增量数据进行聚类具有重要的应用价值。本文提出的基于差分隐私保护的 KDCK-medoids 动态聚类算法，能够采用近邻搜索策略对动态的增量数据进行有效的聚类。针对传统的 K-medoids 算法对初始中心点敏感并且易陷入局部最优等问题，本文提出了 KD 树优化选取中心点算法。由于在动态聚类过程中容易受到外部数据的攻击，该算法也引入了噪声对数据进行扰动，从而达到了隐私保护的目的。然而，在采用 KD 树选取中心点的过程中由于需要计算矩形单元中心和矩形单元密度等，因此其时间损耗比较大。在对增量数据进行动态聚类的同时采用差分隐私保护技术对敏感信息进行保护的相关技术还需要进一步的研究。本文提出的基于差分隐私保护的 KDCK-medoids 动态聚类算法对时间的消耗还比较大，为了今后将其更好地应用于海量数据，还需要对算法进行不断的优化以及进一步的改进。未来的研究主要集中在对算法的优化方面，从而使其更好地应用于海量数据中。

参 考 文 献

[1] 夏宁霞，苏一丹，覃希. 一种高效的 K-medoids 聚类算法[J]. 计算机应用研究，2010，27(12)：4517-4519

[2] Sabzi A，Farjami Y，ZiHayat M. An improved fuzzy k-medoids clustering algorithm with optimized number of clusters[C]// Proceedings of the 11th International Conference on Hybrid Intelligent Systems. IEEE，2011：206-210

[3] 孟颖，罗可，刘建华，等. 一种基于差分演化的 K-medoids 聚类算法[J]. 计算机应用研究，2012，29(5)：1651-1653

[4] Zhu Y T，Wang F Z，Shan X H，et al. K-medoids clustering based on MapReduce and optimal search of medoids[C]// Proceedings of the 9th International Conference on Computer Science and Education. IEEE，2014：573-577

[5] 谢娟英，高瑞. 方差优化初始中心的 K-medoids 聚类算法[J]. 计算机科学与探索，2015，9(8)：973-984

[6] Li T Y，Chen Y，Qu L L，et al. Incremental clustering for categorical data using clustering ensemble[C]// Proceedings of the 29th Chinese Control Conference. IEEE，2010：2519-2524

[7] Srinivas M，Mohan C K. Efficient clustering approach using incremental and hierarchical clustering methods[C]// Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN). IEEE，2010：1-7

[8] Young S，Arel I，Karnowski T P，et al. A Fast and Stable Incremental Clustering Algorithm[C]// Proceedings of the 2010 Seventh International Confe-rence on Information Technology：New Generations (ITNG). IEEE，2010：204-209

[9] Mei J P，Wang Y T，Chen L H，et al. Incremental fuzzy clustering for document categorization [C]// Proceedings of the 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). IEEE，2014：1518-1525

[10] Baili N，Frigui H. Incremental fuzzy clustering with multiple kernels[C]// Proceedings of the 2014 1st International Conference on Advanced Technologies for Signal and Image Processing (ATSIP). IEEE，2014：289-294

[11] 熊平，朱天清，王晓峰. 差分隐私保护及其应用[J]. 计算机学报，2014，37(1)：101-122

[12] 张啸剑，孟小峰. 面向数据发布和分析的隐私保护[J]. 计算机学报，2014，37(4)：927-949

[13] 高小梅，冯云，冯兴杰. 增量式 K-Medoids 聚类算法[J]. 计算机工程，2005，31(1)：181-183

[14] Song Y C，Meng H D，Wang S L，et al. Dynamic and Incremental Clustering Based on Density Reachable[C]// Proceedings of the 2009 Fifth International Joint Conference on INC，IMS and IDC. IEEE，2009：1307-1310

[15] 万静，张义，何云斌，等. 基于 KD-树和 K-means 动态聚类方法研究[J]. 计算机应用研究，2015，32(1)：1-7

[16] Van Rijbergen C J. Information Retrieval (2 nd edition) [M]. London：Butterworths，1979