Journal of Computer Applications
计算机应用,2013,33(9):2419 -2422
ISSN 1001-9081
CODEN JYIIDU
2013-09-01
http://www.joca.cn

文章编号:1001-9081(2013)09-2419-04
doi:10.11772/j.issn.1001-9081.2013.09.2419

基于可变网格划分的密度偏差抽样算法

盛开元*,钱雪忠,吴　秦
(江南大学 物联网工程学院,江苏 无锡 214122)
(*通信作者电子邮箱 shengkaiyuan1991@163.com)

摘　要:简单随机抽样是在分析处理大规模数据集时最常用的数据约简方法,但该方法在处理内部分布不均匀的数据集时容易造成类的丢失。基于固定网格划分的密度偏差抽样算法虽能有效解决该问题,但其速度及效果易受网格划分粒度影响。为此提出了基于可变网格划分的密度偏差抽样算法,根据原始数据集每一维的分布特征确定该维相应的划分粒度,进而构建与原始数据集分布特征一致的网格空间。实验结果表明,在可变网格划分的基础上进行密度偏差抽样,样本质量明显提升,而且相对于基于固定网格划分的密度偏差抽样算法,抽样效率亦有所提高。

关键词:密度偏差抽样;可变网格划分;数据挖掘;大规模数据集;聚类
中图分类号:TP181;TP301.6　　**文献标志码**:A

Density biased sampling algorithm based on variable grid division

SHENG Kaiyuan*, QIAN Xuezhong, WU Qin
(*School of Internet of Things Engineering, Jiangnan University, Wuxi Jiangsu 214122, China*)

Abstract: As the most commonly used method of reducing large-scale datasets, simple random sampling usually causes the loss of some clusters when dealing with unevenly distributed dataset. A density biased sampling algorithm based on grid can solve these defects, but both the efficiency and effect of sampling can be affected by the granularity of grid division. To overcome the shortcoming, a density biased sampling algorithm based on variable grid division was proposed. Every dimension of original dataset was divided according to the corresponding distribution, and the structure of the constructed grid was matched with the distribution of original dataset. The experimental results show that density biased sampling based on variable grid division can achieve higher quality of sample dataset and uses less execution time of sampling compared with the density biased sampling algorithm based on fixed grid division.

Key words: density biased sampling; variable grid division; data mining; large-scale dataset; clustering

0　引言

聚类分析是数据挖掘领域内的重要研究方向之一,但经典的聚类算法仅能在小规模数据集上高效运行,当处理海量、高维的数据集时,运行速度及效果将受影响。解决该问题最有效的方法是对原始数据集进行抽样,即通过对样本数据集聚类分析来推测原始数据集的相关信息[1]。

简单随机抽样(Simple Random Sampling,SRS)是数据挖掘领域内最常用的抽样方法,该方法操作简单且效率较高,但当数据分布不均匀时抽样误差较大[2]。针对这一问题,Palmer等[3]于2000年提出了密度偏差抽样(Density Biased Sampling,DBS)算法,该算法首先将原始数据集划分为不同的组,进而通过建立哈希函数将各组映射到哈希表中,根据各组之间的密度偏差确定各组的抽样概率。相对于SRS算法,DBS算法在处理不均匀数据集时可得到能准确反映原始数据集分布特征的样本数据集,但易受哈希冲突的影响[4]。

近年来针对DBS算法的改进主要围绕数据分组方法展开,如文献[5]中提出的基于树结构的密度偏差抽样算法以及文献[6]中提出的基于网格与树结构的密度偏差抽样算法。以上两种算法虽能有效避免哈希冲突并保证样本质量,但抽样效率较低。2012年,有学者提出了一种基于网格的密度偏差抽样(Grid Density Biased Sampling,G_DBS)算法[7],该算法利用固定的网格结构对原始数据集进行分组,能在相对较短的时间内获得高质量的样本数据集。但如果网格划分粒度过细,抽样效率将降低;如果网格划分粒度过粗,样本质量将受影响。鉴于此,本文在G_DBS算法的基础上,提出了一种基于可变网格划分的密度偏差抽样(Variable Grid Density Biased Sampling,VG_DBS)算法,首先根据原始数据集的分布特征构建特定的网格空间,进而在其基础上执行密度偏差抽样。实验结果表明,相对于G_DBS算法,VG_DBS算法能进一步提高抽样效率并提升样本质量。

1　密度偏差抽样

数据挖掘领域内,密度偏差抽样是一种相对较新的抽样策略,其核心思想是根据原始数据集的分布特征生成样本数据集。实际应用中,首先将原始数据集分成不同的组,各组大小(所含数据点的数量)表示该组的密度,然后按以下原则进行抽样:

1)同一组内各数据点被抽取的概率相等;

2)样本数据集的分布特征与原始数据集一致;

3)各组抽样概率的偏差依据各组大小(密度)的偏差;

4)样本量期望值已知。

当各组大小(密度)之间没有偏差时,密度偏差抽样与简单随机抽样的抽样结果是一致的,因此,简单随机抽样可视为

收稿日期:2013-04-08;**修回日期**:2013-05-08。
基金项目:国家自然科学基金资助项目(61103129,61202312);江苏省科技支撑计划项目(BE2009009)。
作者简介:盛开元(1991-),男,山东临沂人,硕士研究生,主要研究方向:数据挖掘;　钱雪忠(1967-),男,江苏无锡人,副教授,主要研究方向:数据库、数据挖掘、网络计算;　吴秦(1978-),女,江苏宜兴人,副教授,主要研究方向:计算机视觉、模式识别、文本聚类、数据挖掘。

密度偏差抽样的特例。相对于简单随机抽样,密度偏差抽样的优势主要体现在以下两个方面:

1)适应性强。密度偏差抽样过程中,可根据需要确定抽样的核心区域。以对大规模数据集的聚类分析为例,为在包含噪声的数据中发现聚类,可仅对高密度区域抽样;为发现所有聚类,需要既对高密度区域抽样又对低密度区域抽样;为发现离群数据,则需对极低密度区域抽样[8]。

2)约简效果好。由于简单随机抽样是一种等概率抽样方法,因此在高密度区域内会抽取较多的数据点。但在实际应用中,高密度区域内仅需要相对较少的数据点就可以计算出正确结果,对剩余部分继续计算并不会对最终结果有太大影响。密度偏差抽样过程中,不同区域的抽样比例不同,在各区域内单独抽样可产生更为合适的样本。这种抽样方式既保证了样本质量,又在最大限度上缩减了样本实际规模,提高了抽样效率[9]。

2 基于可变网格划分的密度偏差抽样

2.1 固定网格划分

固定网格划分方法基于网格的聚类分析中最常用的网格划分方法,目前也已广泛应用于大规模数据集的聚类分析,多与基于密度的聚类算法相结合[10-11]。固定网格划分是指将数据集的每一维划分成若干个长度相等且互不相交的区间段。对于一个包含 N 个数据点的 d 维数据集 D,其属性$\{a_1,a_2,\cdots,a_d\}$都是有界的,设第 i 维上的值在区间$[l_i,h_i]$中,$i=\{1,2,\cdots,d\}$,则 d 维数据空间可表示为 $S=[l_1,h_1]\times[l_2,h_2]\times\cdots\times[l_d,h_d]$。对于数据集的每一维,采用固定网格划分技术将该维划分成 k 个区间段,则整个数据空间可被划分为 k^d 个网格单元。其中第 i 维上的网格单元的长度 $\theta_i=(h_i-l_i)/k$,该维上第 j 个区间段 $I_{ij}=(l_i+(j-1)\cdot\theta_i,l_i+j\cdot\theta_i)$,$j=\{1,2,\cdots,k\}$[12]。

2.2 可变网格划分

固定网格划分方法通常实现简单,但在高维空间中,会导致计算复杂度呈指数级增加,其可用性也因此而降低[13-14]。不同于固定网格划分方法,可变网格划分方法根据数据的分布特征来划分网格,可大大减少网格单元的数量,而且在构建网格时也更加灵活,目前已有的可变网格划分方法有 GCOD、OptiGrid、MAFIA 等[15]。

本文提出了一种新的可变网格划分方法。简单来讲是指对于原始数据集的每一维,首先对该维进行等深划分,然后通过比较该维各相邻区间段的相似性,对密度相似的相邻区间段执行合并操作。在最终构建的网格空间中,每一维上的区间段个数并不完全相同,而同一维度上各区间段的长度也不完全相同。

设执行合并操作之后第 i 维被划分的区间段个数为 g_i $(i=\{1,2,\cdots,d\})$,则此时的数据空间被分割成 $\prod_{i=1}^{d}g_i$ 个网格单元。而 $g_i\leqslant k$,故 $\prod_{i=1}^{d}g_i\leqslant k^d$。如前文所述,密度偏差抽样需要通过统计各组所含数据点的数量来获取各组密度,因此,网格单元总数越少,统计时的比较次数越少,消耗的总时间也就越少。

可变网格划分方法分为以下四步:

第 1 步 对于第 i 维数据,采用快速排序法排序后的数据可表示为 $D_{ai}=\{q_{i1},q_{i2},\cdots,q_{iN}\}$,$i=\{1,2,\cdots,d\}$。将 D_{ai} 等深划分为 k 个区间段,则各区间段内的数据点个数均为$[N/k]$。此时第 i 维上的第 j 个区间段 $I_{ij}=(q_{i([N/k]\cdot(j-1)+1)},q_{i([N/k]\cdot j)})$,$j=\{1,2,\cdots,k\}$,其中$|I_{ij}|=q_{i([N/k]\cdot j)}-q_{i([N/k]\cdot(j-1)+1)}$。

第 2 步 计算该维各相邻区间段的密度相似性。由于各区间段包含相同数量的数据点,故此处用各区间段的长度$|I_{ij}|$来衡量其密度,并引入参数 ε 定量表示相邻区间段的密度相似性,式(1)给出了 ε 的计算方法。

$$\varepsilon=\begin{cases}|I_{ij}|/|I_{i(j+1)}|, & |I_{ij}|\leqslant|I_{i(j+1)}|\\ |I_{i(j+1)}|/|I_{ij}|, & |I_{ij}|>|I_{i(j+1)}|\end{cases}\tag{1}$$

第 3 步 对于 D_{ai},从第 1 个区间段开始依次比较相邻区间段的密度相似性。如果某两个相邻区间段的密度相似性值 ε 大于阈值 $T(0\leqslant T\leqslant 1)$,表示这两个相邻区间段密度相似。全部比较完成后合并密度相似的相邻区间段。实际应用中,T 的取值越接近于 1,每一维上可合并的相邻区间段相对越少,最终的网格单元总数越多,处理代价相对增加,但能有效保证样本质量;T 的取值越接近于 0,每一维上可合并的相邻区间段相对越多,最终的网格单元总数越少,处理代价相对降低,但样本质量会受到影响。

第 4 步 对数据集的每一维均执行第 1~3 步操作。

2.3 基于可变网格划分的密度偏差抽样策略

密度偏差抽样过程中最关键的步骤是密度的获取。在利用可变网格划分方法构建网格空间时,可通过统计各网格单元内所含数据点的数量来获取各网格单元的密度。另一方面,密度偏差抽样与简单随机抽样最大的区别在于密度偏差抽样是一种非等概率抽样,密度不同的区域,相应的抽样概率也不同。现假设 $\boldsymbol{G}_1$、$\boldsymbol{G}_2$ 为两个网格单元,网格单元密度分别为 n_1、n_2,抽样概率分别为 p_1、p_2。因此,如果 $n_1\neq n_2$,则 $p_1\neq p_2$,故网格单元的抽样概率与其密度之间存在相应的函数关系。

设网格单元 $\boldsymbol{G}_m$ 内所有的数据点为$\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_{n_m}\}$,$m=\{1,2,\cdots,|\boldsymbol{G}|\}$,则该网格单元的密度为 n_m。根据密度偏差抽样的原则,同一网格单元内各数据点 $\boldsymbol{x}$ 被抽取的概率 $P(\boldsymbol{x}|\boldsymbol{x}\in\boldsymbol{G}_m)$ 是相等的,如上文所述,该概率是关于 n_m 的函数 $f(n_m)$,即 $P(\boldsymbol{x}|\boldsymbol{x}\in\boldsymbol{G}_m)=f(n_m)$。$f(n_m)$ 的函数表达式如下:

$$f(n_m)=a/n_m^e;\ 0\leqslant e\leqslant 1\tag{2}$$

其中 e 为常量。当 $e=0$ 时,各网格单元的抽样概率函数相同,抽样结果与简单随机抽样结果相同;当 $e=1$ 时,在各网格单元内将抽取相同数量的数据点。

密度偏差抽样过程中,样本总量 n 是每个网格单元中所抽取样本量的总和,因此

$$n=\sum_{m=1}^{|\boldsymbol{G}|}n_m f(n_m)=\sum_{m=1}^{|\boldsymbol{G}|}n_m\frac{a}{n_m^e}\tag{3}$$

其中$|\boldsymbol{G}|$为网格单元总数。由式(3)可推出:

$$a=n\Big/\sum_{m=1}^{|\boldsymbol{G}|}n_m^{1-e}\tag{4}$$

因此,由式(2)~(4)可推出每个网格单元的抽样概率函数为:

$$f(n_m)=n\Big/\Big(n_m^e\sum_{m=1}^{|\boldsymbol{G}|}n_m^{1-e}\Big)\tag{5}$$

2.4 基于可变网格划分的密度偏差抽样算法

输入 原始数据集 $\boldsymbol{D}$,样本量期望值 n,参数 e、T。

输出 样本数据集 $\boldsymbol{SD}$。

第 1 步 扫描原始数据集,用可变网格划分方法构建网格空间 $\boldsymbol{G}$,网格单元总数为$|\boldsymbol{G}|$,并统计各网格单元内数据点的个数 n_m,$m=\{1,2,\cdots,|\boldsymbol{G}|\}$。

第 2 步 根据式(4)计算 a 的值。

第 3 步 对于第 m 个网格单元 $\boldsymbol{G}_m$,如果 $n_m\neq 0$,根据式(2)计算 $f(n_m)$;否则令 $m=m+1$。

第 4 步 在网格单元 $\boldsymbol{G}_m$ 上执行简单随机抽样,获得 $n_m\cdot f(n_m)$ 个样本数据点。

第 5 步　对每一个网格单元均执行第 3 ~ 4 步操作。

第 6 步　输出样本数据集 ***SD***。

实验过程中,参数 e、T 的取值分别为 0.5,0.6。

3　实验与分析

为验证 VG_DBS 算法的可行性,通过实验分别对 VG_DBS 算法、G_DBS 算法和 SRS 算法在抽样效果、样本质量和执行时间三个方面进行比较分析。实验在同一台 PC 机(CPU 2.50 GHz,内存 2.00 GB)上进行,操作系统版本为 Windows XP,实验工具为 Matlab 7.0。实验选用 1 组人工数据集 Dataset 和 5 组 UCI 标准数据集:clean、page-blocks、shuttle、ConfLongDemo_JSI、poker-hand。各数据集的详细信息如表 1 所示。

3.1　抽样效果对比分析

本组实验选用二维人工数据集 Dataset,对其按 1% 的比例抽取样本的结果如图 1 所示,其中图 1(b) ~ (d)依次为 SRS 算法、G_DBS 算法和 VG_DBS 算法的抽样结果。

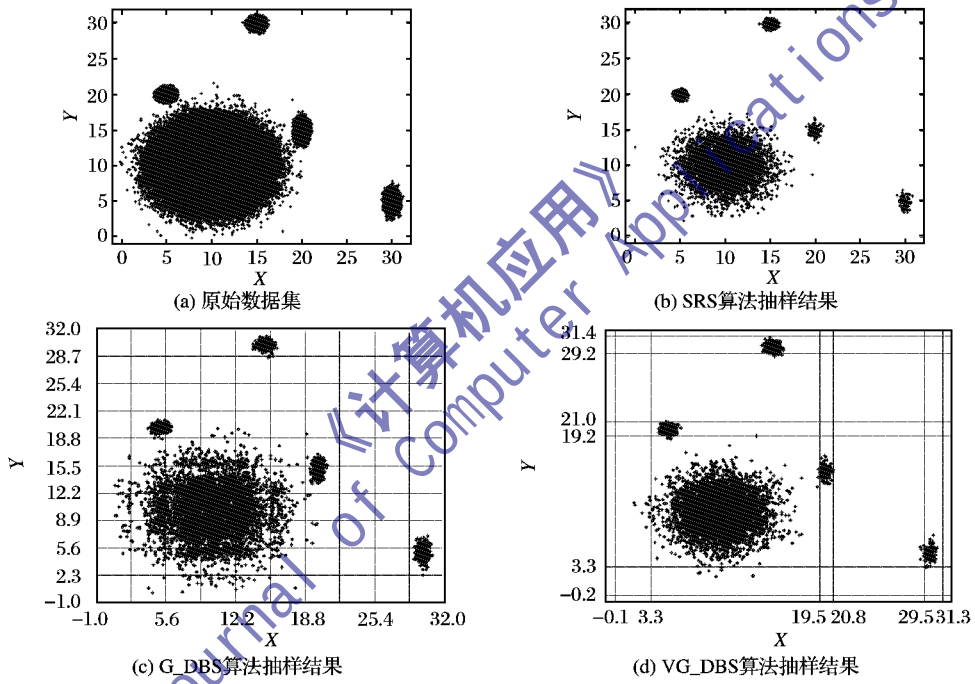

图 1　各算法抽样效果对比

1)由图 1(b) ~ (d)可看出,VG_DBS 算法在低密度区域内所抽取的样本量要明显多于 SRS 算法,有效保证了样本数据分布与原始数据分布的一致性。

2)由图 1(c) ~ (d)可看出:相对于 G_DBS 算法,VG_DBS 算法在抽样过程中所构建的网格单元总数要远远少于 G_DBS 算法,大大减少了数据统计所需时间;并且,VG_DBS 算法在高密度区域内所抽取的样本量要少于 G_DBS 算法,有效增强了样本的约简效果。

表 1　数据集信息

数据集名称	数据量	维度	聚类个数	最大类数据量	最小类数据量
Dataset	720 000	2	5	500 000	10 000
clean	476	166	2	267	209
page-blocks	5 473	10	5	4 913	28
shuttle	43 500	9	7	34 108	6
ConfLong-Demo_JSI	164 860	5	11	54 480	1 381
poker-hand	1 025 010	10	10	513 701	8

3.2　样本质量对比分析

样本质量包括样本完整性和样本正确性两个方面,样本完整性是指样本数据集中包含的聚类个数是否与原始数据集一致,样本正确性是指在样本数据集上进行聚类分析的结果是否正确。为保证实验结果的客观性,各实验中 VG_DBS 算法在对原始数据集进行可变网格划分时,每一维初始划分的区间段个数与 G_DBS 算法在每一维所划分的区间段个数一致。

首先,对各算法所抽取样本的完整性进行对比分析,实验采用内部分布不均匀、偏斜较大的数据集 shuttle。该数据集中,最大类包含 34 108 条数据,最小类仅包含 6 条数据。现分别采用 VG_DBS 算法、G_DBS 算法和 SRS 算法对该数据集在不同比例下进行抽样,记录各抽样算法所抽取的样本中所包含的聚类个数,实验结果如表 2 所示。表 2 中的实验数据表明:针对 shuttle 数据集,相对于 SRS 算法,VG_DBS 算法和 G_DBS 算法在抽样比例为 5% 时就不会丢失类,而 SRS 算法则需要至少 20% 的抽样比例;相对于 G_DBS 算法,VG_DBS 虽然减少了网格单元总数,但在抽样时并没有丢失类,因此算法在增强了约简效果的基础上保证了样本的完整性。

表 2　各算法在 shuttle 数据集上的抽样测试结果

抽样比例/%	正确结果	VG_DBS	G_DBS	SRS
1	7	6	6	5
5	7	7	7	5
10	7	7	7	6
15	7	7	7	6
20	7	7	7	7

接下来对各算法所抽取样本的正确性进行对比分析。聚类算法采用传统的最大期望(Expectation Maximization,EM)算法,聚类结果的正确率定义为被正确聚类的数据量与样本量之比。各抽样算法在不同数据集上的实验结果如表 3 所示。表 3 中的实验数据表明:针对实验中所选用的 5 组数据集,在保证三种抽样算法均不丢失类的前提下,采用相同的抽样比例进行抽样后,VG_DBS 算法所得到的样本的正确率普遍高于另外两种算法。

3.3 执行时间对比分析

首先对各抽样算法在人工数据集 Dataset 上的执行时间进行对比分析。通过改变 Dataset 的数据量，分别测试 VG_DBS 算法、G_DBS 算法和 SRS 算法抽取 1% 样本所需时间以及样本聚类时间，实验结果如表 4 所示。

表 4 中的实验数据表明：针对人工数据集 Dataset，在获得等量样本时，SRS 算法执行时间最少，VG_DBS 算法执行时间明显少于 G_DBS 算法；在对样本进行聚类时，VG_DBS 算法所抽取样本的执行时间普遍低于另外两种算法；而且随着样本量的逐渐增加，三种算法在执行时间上的差异也逐渐增大。

其次对各抽样算法在 UCI 标准数据集上的执行时间进行对比分析，在保证三种抽样算法均不丢失类的前提下，分别测试 VG_DBS 算法、G_DBS 算法和 SRS 算法抽取相同数量的样本所需时间以及样本聚类时间，抽样比例同表 3，实验结果如表 5 所示。表 5 中的实验数据表明：针对 UCI 标准数据集，VG_DBS 算法同样具有一定的优势。首先，虽然在数据规模较小时，VG_DBS 算法在执行时间上并无明显优势（以 clean 数据集为例），但当数据规模增大时，VG_DBS 算法的执行时间要远远少于 G_DBS 算法，而且样本质量也有所提高；其次，VG_DBS 算法的抽样时间要远远多于 SRS 算法，但随着数据规模的逐渐增大，VG_DBS 算法所抽取样本在聚类时所消耗的时间要少于 SRS 算法。而且基于 VG_DBS 算法的聚类结果的正确性也高于基于 SRS 算法和 G_DBS 算法的聚类结果，这充分体现了 VG_DBS 算法的可行性。

表 3 样本正确性测试结果 %

抽样算法	clean		page-blocks		ConfLongDemo_JSI		Dataset		poker-hand	
	抽样比例	正确率	抽样比例	正确率	抽样比例	正确率	抽样比例	正确率	抽样比例	正确率
VG_DBS	10	95.38	1	96.28	2	93.31	0.1	99.86	10	92.54
G_DBS	10	95.16	1	95.42	2	92.43	0.1	95.57	10	81.78
SRS	10	75.00	1	83.64	2	81.63	0.1	98.97	10	82.77

表 4 各算法在 Dataset 数据集上的执行时间 s

抽样算法	数据量 180 000		数据量 360 000		数据量 540 000		数据量 720 000		数据量 900 000	
	抽样时间	聚类时间	抽样时间	聚类时间	抽样时间	聚类时间	抽样时间	聚类时间	抽样时间	聚类时间
VG_DBS	0.250	0.797	0.453	0.375	0.610	0.890	0.828	0.766	0.984	1.391
G_DBS	0.422	0.921	0.782	1.484	1.094	1.609	1.454	0.750	1.844	3.641
SRS	0.032	0.484	0.016	0.593	0.031	2.656	0.015	0.735	0.031	9.704

表 5 各算法在 UCI 标准数据集上的执行时间 s

抽样算法	clean		page-blocks		ConfLongDemo_JSI		poker-hand	
	抽样时间	聚类时间	抽样时间	聚类时间	抽样时间	聚类时间	抽样时间	聚类时间
VG_DBS	2.969	11.406	28.203	2.922	26.281	16.515	212.184	718.075
G_DBS	1.468	11.328	53.984	7.312	440.829	24.843	2658.465	775.242
SRS	0.016	1.140	0.016	0.094	0.016	24.765	0.074	758.298

4 结语

本文提出了一种基于可变网格划分的密度偏差抽样算法，相对于已有的基于网格的密度偏差抽样算法，该算法最大的特点是能够针对特定数据集构建一个符合该数据集分布特征的网格，较好地解决了大规模不均匀数据集的抽样问题。相对于简单随机抽样算法，该算法所抽取样本的质量有显著提升；而相对于基于固定网格划分的密度偏差抽样算法，该算法在保证样本质量的同时降低了抽样过程的执行时间。但是，本文的研究工作还有待进一步深入和扩展，如对海量高维数据集的处理时间有待进一步缩短等。

参考文献：

[1] 张春阳，周继恩，钱权，等. 抽样在数据挖掘中的应用研究[J]. 计算机科学，2004，31(2)：126－128.

[2] GU B H，HU F F，LIU H. Sampling and its application in data mining：a survey[R]. Singapore：National University of Singapore，2000.

[3] PALMER C R，FALOUTSOS C. Density biased sampling：an improved method for data mining and clustering[C]// Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. New York：ACM Press，2000：82－92.

[4] 胡文瑜，孙志辉，吴英杰. 数据挖掘取样方法研究[J]. 计算机研究与发展，2011，48(1)：45－54.

[5] NANOPOULOS A，THEODORIDS Y，MANOLOPOULOS Y. Indexed-based density biased sampling for clustering applications[J]. Data & Knowledge Engineering，2006，57(1)：37－63.

[6] APPEL A P，PATERLINI A A，de SOUSA E P M，*et al*. A density-biased sampling technique to improve cluster representativeness[C]// Proceedings of PKDD 2007. Berlin：Springer，2007：366－373.

[7] HUANG J B，SUN H L，KANG J M，*et al*. ESC：an efficient synchronization-based clustering algorithm[J]. Knowledge-Based Systems，2013，40：111－122.

[8] 唐成龙，邢长征. 基于数据分区和网格的离群点挖掘算法[J]. 计算机应用，2012，32(8)：2193－2197.

[9] 余波，朱东华，刘嵩，等. 密度偏差抽样技术在聚类算法中的应用研究[J]. 计算机科学，2009，36(2)：207－209.

[10] ZHAO Y C，CAO J，ZHANG C Q，*et al*. Enhancing grid-density based clustering for high dimensional data[J]. Journal of Systems and Software，2011，84(9)：1524－1539.

[11] PILEVAR A H，SUKUMAR M. GCHL：a grid-clustering algorithm for high-dimensional very large spatial data bases[J]. Pattern Recognition Letters，2005，26(7)：999－1010.

[12] 张建锦，吴渝，刘小霞. 一种改进的密度偏差抽样算法[J]. 计算机应用，2007，27(7)：1695－1698.

[13] 贺玲，蔡益朝，杨征. 高维数据的相似性度量研究[J]. 计算机科学，2010，37(5)：155－156.

[14] 贺玲，蔡益朝，杨征. 高维数据空间的一种网格划分方法[J]. 计算机工程与应用，2011，47(5)：152－153.

[15] 赵卓真. 一种基于密度与网格的聚类算法[D]. 广州：中山大学，2012.