基于折叠技术的大数据样本洗牌算法研究

李 庆,刘涵阅,张春生*
(内蒙古民族大学 计算机科学与技术学院,内蒙古 通辽 028043)

摘 要:大数据处理效率问题是目前的研究热点,而基于样本抽样技术可降样本数量,是提高大数据处理效率的方法之一。文中提出一种基于折叠技术的大数据洗牌算法,首先给出洗牌算法的基本原理,同时定义离散度和均匀度两个评价指标,并从时间效率、离散度和均匀度 3 个角度进行了仿真实验。实验结果表明,基于折叠技术的大数据洗牌算法具有较高的时间效率,当样本分段数为样本总数的 5%,循环次数为样本总数的 2% 时,离散度和均匀度明显优于其他基于随机技术的洗牌算法。基于折叠技术的大数据洗牌算法为大数据抽样和提高局部样本的可用性提供了一个新的途径,克服了抽样不均匀对原始样本产生的影响,提高了大数据挖掘的时间效率。

关键词:折叠技术;大数据;洗牌算法;局部有效性;Guid
中图分类号:TP391 **文献标识码**:A **文章编号**:1673-629X(2021)05-0043-05
doi:10.3969/j.issn.1673-629X.2021.05.008

Research on Big Data Sampling Shuffle Algorithm Based on Folding Technology

LI Qing, LIU Han-yue, ZHANG Chun-sheng*
(School of Computer Science and Technology, Inner Mongolia University for Nationalities, Tongliao 028043, China)

Abstract: The efficiency of big data processing is currently a hot research topic. The technology based on sample sampling can reduce the number of samples, which is one of the methods to improve the efficiency of big data processing. A big data sampling shuffling algorithm based on folding technology is proposed. Firstly, the principle of the shuffling algorithm is given, and two indexes of dispersion and uniformity are defined. Simulation experiments are carried out from three angles of time efficiency, dispersion and uniformity. The experiment shows that the shuffling algorithm based on folding technology has comparative high time efficiency. When the number of sample segments is 5% of the total number of samples, and the number of cycles is 2% of the total number of samples, the dispersion and uniformity are significantly over-performed than other shuffling algorithms based on random techniques. Big data shuffling algorithm based on folding technology provides a new way for big data sampling and improving the availability of local samples, which overcomes the influence of uneven sampling on the original samples and improves the time efficiency of big data mining.

Key words: folding technology; big data; shuffling algorithm; local effectiveness; Guid

0 引 言

大数据分析是目前研究和应用的热点,近几年在大数据分析领域的研究取得了长足的发展,但大数据分析的效率问题仍然是发展的瓶颈。舍恩伯格和库克耶指出:大数据不用随机分析法这样的捷径,而采用所有数据的方法。所谓“所有数据”是一种相对的说法,但在问题思路上,似乎又回转向了“全面调查”,数据科学家甚至提出了“样本=总体”的准则。

对“样本=总体”的观点存在争议,事实上不可能完全利用存在无效信息的全部大数据进行分析,因此采样调查仍然具有可行性。采样调查强调的是“窥一斑而知全豹”,从充分均匀的样本中选取一部分,就能有效推断总体的情况[1-6]。

但在大数据时代,面对大量的数据及源源不断的数据流,如何科学地从中选取合适的样本,从而达到保证采样调查样本的精度和统计分析的目的,这是大数据时代下采样调查面临的最大问题。另外,采样后的局部数据是否能真实反映全局数据的规则也是探讨和研究的一个重要课题。一个潜在的解决方案是给出近似结果,也就是由抽样产生的局部数据生成的隐知识

收稿日期:2020-05-09 **修回日期**:2020-09-11
基金项目:国家自然科学基金(81460656);内蒙古自然科学基金(2018MS06016)
作者简介:李 庆(1996-),男,硕士研究生,研究方向为数据挖掘;通信作者:张春生(1965-),男,教授,硕导,研究方向为数据库技术、数据挖掘、大数据分析与处理。

来近似表示全局的隐知识。

得到正确可用的局部数据的前提是要有一个好的大数据洗牌算法,鉴于随机抽样算法存在样本分布不够理想的现实[7-11],该文首先提出一种基于折叠技术的洗牌算法。该算法来源于生活中的扑克洗牌原理,算法简单易行,不受时间种子的影响,具有较高的时间效率、离散度和均匀度。基于折叠技术的大数据洗牌算法为大数据抽样和提高局本样本的可用性提供了一个新途径。

1 基于随机序列的洗牌算法

为了与该文提出的基于折叠洗牌技术的大数据抽样算法进行对比,采用目前比较流行的 2 种不重复随机采样算法,即基于哈希技术和基于 Guid 技术的不重复随机采样算法。

1.1 基于哈希技术的洗牌算法

利用哈希表来生成无冲突序列算法的基本原理是[12-14],首先定义一个空哈希表,通过随机函数 Rand()生成一个随机数,并判断哈希表里是否有与之相同的随机数,如果有则重新调用 Rand()函数,如果没有,则将该随机数放入哈希表,并使其关键码值也等于该随机数。由于该序列的每一个关键码值与其所对应的数据值相等,所以可以直接通过关键码的映射进行按值查询。

算法如下:

```
Hashtable hashtable=new Hashtable();
Rand() rm=new Rand() ;
int RmNum=N;//N 为随机数个数
for (int i=0;hashtable.Count<RmNum;i++)
{
int nValue=rm.Next(100);
if(! hashtable.ContainsValue(nValue) && nValue! =0)
  {
   hashtable.Add(nValue, nValue);
  }
}
```

1.2 基于 Guid 技术的洗牌算法

Guid 又称为全局唯一标识符[15],在理想情况下,任何计算机和计算机集群都不会生成两个相同的 Guid 值,一般表示成 32 个 16 进制数字(0-9,A-F)组成的字符串,它实质上是一个 128 位长的二进制整数。

算法首先定义一个空序列,并调用 Guid 方法生成一个不唯一的数,然后将这个数作为随机种子放入 Rand()函数中得到一个随机数,接着将这个随机数放入刚才定义的空序列,重复以上操作,最终会得到一个随机序列。

算法如下:

```
private void btn_jdsjxp_Click(object sender, EventArgs e)
{
int i_ybzs; //样本总数
int i;  //设置样本循环变量
int k;  //随机下标
i_ybzs=int.Parse(tb_ybs.Text);
//样本总数转换为整
int[] yb_s=new int[i_ybzs];
//定义原始样本序列
int[] yb_d=new int[i_ybzs];
//定义目标样本序列
for(i=0;i<i_ybzs;i++)
//初始化样本序列
{
yb_s[i]=i+1;
yb_d[i]=0;
}
for(i=0;i<i_ybzs;i++)
//开始对所有样本循环
{
k=GetRandNumber(0, i_ybzs-1);
//随机选择不重复样本下标
yb_d[i]=yb_s[k];
}
```

2 基于折叠技术的洗牌算法

2.1 基于折叠技术的洗牌算法的优势

鉴于随机抽样算法受到时间种子的影响,采样分布不够均匀,而折叠洗牌算法模仿扑克牌的洗牌原理,进行多次分段均匀组合,算法的分布不受随机数限制,全程均匀分布,同时由于不进行重复数判断,所以,无论从数据分布还是时间效率上都应比随机抽样算法优越。

2.2 折叠技术的洗牌算法描述

基于折叠洗牌技术的采样算法基本原理是,折叠洗牌算法模拟扑克的洗牌过程,设样本总数为 $n*k+p$,其中 p 和 k 代表段长,$p \leqslant k$。当 $p=k$ 时,$n*k+p=(n+1)*k$,数据分 $n+1$ 段;当 $p<k$ 时,数据分 n 段,另有一个不足 k 长的长度为 p 的尾段。在洗牌过程中 k 长数据分段在折叠过程中可以头头连接,也可以根据随机数进行头头、头尾、尾尾连接。不足 k 长度的样本可不参与折叠,直接加到序列尾部。

例如 $n+1$ 段 k 长样本段的头头连接方法如下:

$I_{11},I_{12},\cdots,I_{1k}$

$I_{21},I_{22},\cdots,I_{2k}$

$\cdots$

$I_{n1},I_{n2},\cdots,I_{nk}$

$I_{(n+1)1},I_{(n+1)2},\cdots,I_{(n+1)k}$

头头连接为:

$I_{11},I_{21},\cdots,I_{n1},I_{(n+1)1},I_{12},I_{22},\cdots,I_{n2},I_{(n+1)2},\cdots,I_{1k},I_{2k},\cdots,I_{nk},I_{(n+1)k}$

若存在不足 k 长的 p 长子段 $I_{(n+1)1},I_{(n+1)2},\cdots,I_{(n+1)p}$,则直接加到序列尾部。

头头连接为:

$I_{11},I_{21},\cdots,I_{n1},I_{12},I_{22},\cdots,I_{n2},\cdots,I_{1k},I_{2k},\cdots,I_{nk},I_{(n+1)1},I_{(n+1)2},\cdots,I_{(n+1)p}$

该文认为折叠洗牌算法不受时间种子的影响,均匀度和离散度高于随机数算法,时间效率高于随机数算法。

2.3　经典洗牌算法与折叠技术的洗牌算法时间效率分析

哈希算法在生成随机序列的时候,每生成一个随机数之前,都会进行一次冲突检测,假设当前检测的序列长度为 n,那么每一次检测所消耗的时间平均量为 $n/2$。如果要保证每次添加的随机数都不重复,则需要做 n 次检测,其时间复杂度为:

$$T(n)=\sum_{i=0}^{n-1}\frac{i}{2}=\frac{n^2-n}{2} \tag{1}$$

用大 O 法表示即为 $O(n^2)$。

Guid 算法的核心在于用微软的 Guid 标准生成一个全球唯一的 128 位数字,并将其作为 Rand( )函数的种子,来生成一个不重复的数。由于 Guid 属于微软内部实现的功能,这里无法对其时间复杂度进行直接评价,于是将其所在函数 GetRandNumber( )的时间复杂度记为 m。那么整体算法的时间复杂度可以视为:

$$O(n)=n+mn \tag{2}$$

而基于折叠技术的洗牌算法由于只是将原始数据序列分割成 n 段,有 n 段重新组合生成新的目标序列,所以其总体时间复杂度为:

$$O(n)=n \tag{3}$$

显然,相比前两种经典的算法,从理论上看,基于折叠技术的洗牌算法的时间复杂度更小,运行速度相对更快,效率更高。

3　洗牌算法评价因子

均匀度和离散度是衡量抽样算法数据分布的 2 个指标。

设样本分成 n 段,每段长度为 k。

3.1　均匀度

设:

$$\mathrm{sum}(i)=\sum_{m=1}^{k}I_{im}$$

$$\text{均匀度}=\frac{\sum_{i=2}^{n}(\mathrm{sum}(i)-\mathrm{sum}(i-1))}{n-1}$$,代表了相邻分段数据间的相异程度,均匀度越小,效果越好。

3.2　离散度

设有 n 个样本:$I_1,I_2,\cdots,I_n$

$$\text{离散度}=\frac{\sum_{i=2}^{n}(I_i-I_{i-1})}{n-1}$$,代表了相邻数据间的相异程度,离散度越大,效果越好。

4　仿真实验

该文对上面提到的 3 种洗牌算法从时间效率、均匀度、离散度进行比较,从而证明基于折叠技术的洗牌算法的优越性。

4.1　数据准备

应用 C#语言开发出实验程序,实验系统设置样本总数、最大分割段数、循环次数和均匀度及离散度分析时的分段数。在折叠方式可采用单向折叠和随机双向折叠,根据系统产生的随机数决定每个分段的折叠方向。同时在最大分段数的范围内,可采用固定分段和随机分段的方式进行,通过各项功能的设置,提高了实验程序的灵活性,满足不同的实验需要。

4.2　实验过程与结果

(1)算法时间效率对比分析。

对 Hash 算法、Guid 算法、折叠技术 3 种洗牌算法进行时间效率对比分析,样本数量从 1 000 到 10 000,增量为 1 000,对比结果如表 1 所示,对比图如图 1 所示。

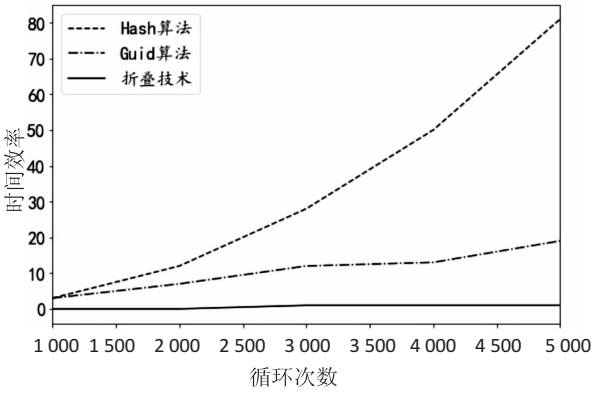

图 1　算法时间效率分析

表 1　算法时间效率分析

算法	1 000	2 000	3 000	4 000	5 000	6 000	7 000	8 000	9 000	10 000
Hash 算法	3	12	28	50	81	116	158	206	263	324
Guid 算法	3	7	12	13	19	23	26	29	32	36
折叠技术	0	0	1	1	1	2	2	2	3	3

(2)离散度对比分析。

对 Hash 算法、Guid 算法、折叠技术 3 种洗牌算法进行离散度对比分析,样本数量为 1 000,分段数量为 50 段,循环次数从 10 到 50,对比结果如表 2 所示,对比图如图 2 所示。

表 2 基于循环次数变化的离散度对比

算法	10	20	30	40	50
Hash 算法	328	336	356	328	321
Guid 算法	336	341	341	340	327
折叠技术	374	491	460	268	75

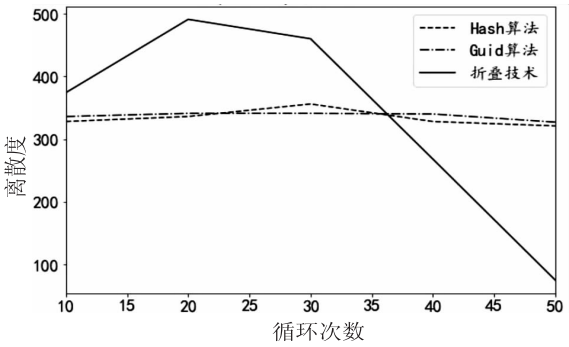

图 2 基于循环次数变化的离散度对比

对 Hash 算法、Guid 算法、折叠技术 3 种洗牌算法进行离散度对比分析,样本数量为 1 000,分段数量为 10 ~ 50 段,循环次数 20,对比结果如表 3 所示,对比图如图 3 所示。

表 3 基于分段数变化的离散度对比

算法	10	20	30	40	50
Hash 算法	330	324	324	329	333
Guid 算法	329	336	328	331	336
折叠技术	18	30	111	468	491

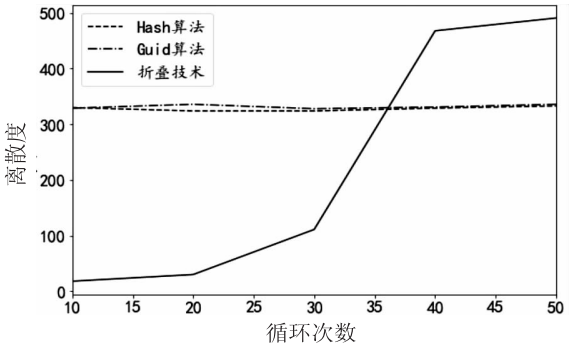

图 3 基于分段数变化的离散度对比

(3)均匀度对比分析

对 Hash 算法、Guid 算法、折叠技术 3 种洗牌算法进行离散度对比分析,样本数量为 1 000,分段数量为 50 段,循环次数从 10 到 50,对比结果如表 4 所示,对比图如图 4 所示。

表 4 基于循环次数变化的均匀度对比

算法	10	20	30	40	50
Hash 算法	1 098	1 242	1 266	1 360	1 395
Guid 算法	1 599	1 587	1 096	1 079	1 444
折叠技术	197	978	552	482	1 538

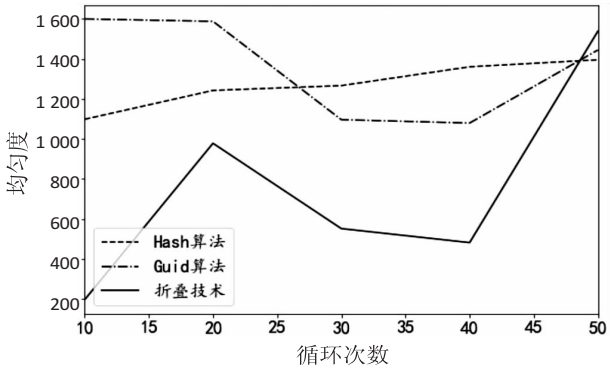

图 4 基于循环次数变化的均匀度对比

对 Hash 算法、Guid 算法、折叠技术 3 种洗牌算法进行离散度对比分析,样本数量为 1 000,分段数量为 10 ~ 50 段,循环次数 20,对比结果如表 5 所示,对比图如图 5 所示。

表 5 基于分段数变化的均匀度对比

算法	10	20	30	40	50
Hash 算法	4 390	2 268	1 742	1 849	1 259
Guid 算法	3 287	2 325	1 697	1 651	1 531
折叠技术	88	3548	585	678	978

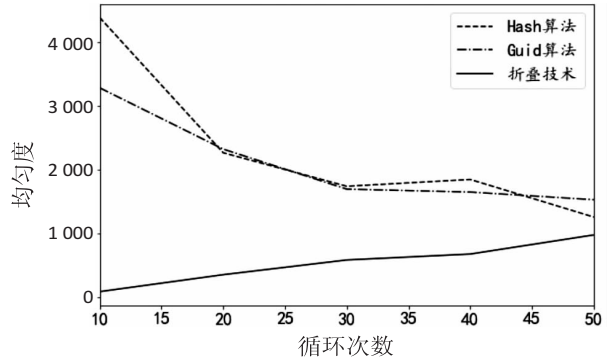

图 5 基于分段数变化的均匀度对比

4.3 结果分析

(1)算法时间效率对比分析。

从实验结果来看,折叠洗牌算法从时间效率来看远远优于 Hash 算法和 Guid 算法,这与理论分析一致,从而证明折叠洗牌算法具有时间效率优越性。

(2)离散度分析。

从离散度因子定义来看,离散度越大,说明数据离散的好,实验结果表明,当分段数大于等于 40,循环次数小于等于 30 时,折叠洗牌算法具有明显的优势,同时也看到分段数与循环次数的变化对 Hash 算法和 Guid 算法的离散度改变不大,几乎没有影响。

(3)均匀度分析。

从均匀度因子定义来看,均匀度越小,说明数据离散的好,实验结果表明,当分段数小于等于 50,循环次数小于等于 40 时,折叠洗牌算法具有明显的优势。

(4)综合评价。

从时间效率来看,折叠洗牌算法远远优于 Hash 算法和 Guid 算法;综合离散度和均匀度 2 因素,当分段数在[40,50]区间,循环次数在[10,30]区间时,折叠洗牌算法具有非常好的效果。同时也要注意到:分段数与循环次数的变化对 Hash 算法和 Guid 算法的离散度几乎没有影响,这也是基于随机技术的致命缺点。

通过实验表明,当样本数为 1 000,分段数为 50,循环次数为 20 时,效果最佳,也就是当分段数为样本总数的 5%,循环次数为样本总数的 2% 时,达到最佳效果。

5　结束语

提高大数据处理效率问题是大数据研究的热点,成熟的方案也很多,但基于抽样技术的大数据处理方法不仅适合于静态数据处理,也适合流式数据处理。一个好的大数据洗牌算法能保证抽样样本的可用性。该文从生活中的扑克洗牌算法得到启示,提出一种大数据洗牌算法,算法原理简单,易于实现,从实验结果来看,当样本分段数为样本总数的 5%,循环次数为样本总数的 2% 时,具有最佳效果,明显优于其他基于随机技术的常规算法。

参考文献:

[1] 邱　东. 大数据时代对统计学的挑战[J]. 统计研究,2014,31(1):16-22.

[2] MAYER-SCHONBERGER V,CUKIER K. Big data:a revolution that will trans-form how we live,work,and think[M]. [s. l.]:John Murray Publishers,2013.

[3] 徐建军,张国华. 基于 Apriori 数据挖掘算法的应用与实践[J]. 计算机技术与发展,2020,30(4):206-210.

[4] 吴　颖,李晓玲,唐晶磊. Hadoop 平台下粒子滤波结合改进 ABC 算法的 IoT 大数据特征选择方法[J]. 计算机应用研究,2019,36(11):3297-3301.

[5] 汪　洋,李　鹏,季一木,等. 高性能计算与天文大数据研究综述[J]. 计算机科学,2020,47(1):1-6.

[6] 杨国强,丁杭超,邹　静,等. 基于高性能密码实现的大数据安全方案[J]. 计算机研究与发展,2019,56(10):2207-2215.

[7] 王英强,张卫钢,王红刚. 基于 NB-IoT 的农业数据采集系统的设计[J]. 计算机技术与发展,2020,30(2):206-210.

[8] WANG Xingyuan, ZHANG Huili. A color image encryption with heterogeneous bit-permutation and correlated chaos[J]. Optics Communications,2015,342:51-60.

[9] 韩露露,杨　波,来齐齐,等. 一种组合式伪随机数发生器的构造[J]. 小型微型计算机系统,2019,40(3):573-578.

[10] 朱淑芹,王文宏,李俊青,等. 一类二次多项式混沌及其随机数生成器设计[J]. 计算机工程与应用,2018,54(9):84-88.

[11] 马　上,刘剑锋,杨泽国,等. 基于余数系统与置换多项式的高速长周期伪随机序列生成方法[J]. 电子与信息学报,2018,40(1):42-49.

[12] DEVINE R. Design and implementation of DDH:a distributed dynamic hashing algorithm[M]//Foundations of data organization and algorithms. Berlin:Springer,1993:101-114.

[13] 张利华. 基于随机数和 Hash 函数的认证方案[J]. 微电子学与计算机,2007,24(6):80-83.

[14] EHDAIE M, ALEXIOU N, AHMADIAN M, et al. 2D hash chain robust random key distribution scheme[J]. Information Processing Letters,2016,116(5):367-372.

[15] FERREIRA R, AGUIAR R, MATOS A. Recognizing entities across protocols with unified UUID discovery and asymmetric keys[C]//GLOBECOM IEEE global communications conference (USA). Atlanta,GA,USA:IEEE,2013.