二维混合数据分布下相关性检测的新方法 HY-COCA

曹 巍 王秋月 覃雄派 王 珊
(中国人民大学信息学院 北京 100872)

摘 要 混合数据分布是指数据分布的不同区域具有不同的特殊分布。例如销售额和地区两个属性之间,在销售额比较低的数值区间中,两者呈现近似相互独立的数据分布;而在销售额比较高的数值区间,二者呈现近似函数依赖的数据分布。现有检测数据相关性的研究专注于给出一个总体的二维相关性的度量,而无法检测出子区域的特殊相关性。在统计分析时,这类具有特殊相关性的子区域有更丰富的统计意义,值得引起重视。研究并提出了存在这类混合数据分布的情况下,检测数据相关性的新方法 HY-COCA。该方法在熵相关系数的基础上,缩小了子区域的搜索空间,与 Naive 方法相比,降低了复杂度;同时 HY-COCA 还讨论了子区域的相关性差异判别与结果展示等问题。在生成的数据和测试基准数据上进行了实验,结果验证了方法的有效性。

关键词 数据分布,混合数据分布,相关性,数据分布区域,相关性差异分数
中图法分类号 TP311.13 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.6.042

HY-COCA: A Hybrid-data-distribution-aware Way to Detect Correlation over Bi-dimensional Data Space

CAO Wei WANG Qiu-yue QIN Xiong-pai WANG Shan
(School of Information, Renmin University of China, Beijing 100872, China)

Abstract Hybrid data distribution between two attributes means that different data sub-regions exhibit different correlated associations. For example, in a distribution between sale amounts and different cities, a semi-independent distribution is observed with lower sale amounts, but for higher sale amounts, the two attributes present soft functional dependency. Previous researches on auto detection of association focused on deducing an overall measure of association over two dimensional distributions. They were unable to address hybrid data distribution problem. In statistical analysis, such sub-regions with particular data associations are worth paying attention to. This paper proposed a new way, HY-COCA, to detect data associations globally and locally, finding those sub-regions with special data associations. We did experiments on both synthetic and benchmark data. Experimental results verify the effectiveness of HY-COCA.

Keywords Data distribution, Hybrid data distribution, Data association, Sub-regions in data distribution, Differentiating score of association

1 引言

在数据管理技术中,数据的相关性信息是非常有价值的统计信息,对于关系数据库查询优化统计值的估计、数据挖掘、生物信息分析等应用很有意义。

在关系数据库查询优化时,为两项合取谓词的选择操作估计结果选择率有两种方法,一种是假定合取谓词的两列属性值相互独立,另一种是认为两列属性值具有关联性,因此可以利用二维直方图支持选择率的估计。但是可以看出,这两种假设都没有定量的依据,如果数据间本身有相关性,假设属性值相互独立,会造成估计结果数有很大的误差;如果数据间的相关性很弱,则创立和维护二维直方图均须付出一定的时间和空间代价。因此文献[3]给出了根据描述数据相关性的定量指标——数据相关系数,来判断不同类型的数据分布,根据不同的数据分布[1]创建不同的直方图的方法 COCA-Hist。判断数据间的相关性是数据管理和数据分析应用首先要面对的问题。目前的研究大多停止于此。但是针对现实数据,只给出在总体分布上的一个笼统的数据间的相关性判断或者度量是不够的[2]。比如一家公司全部的员工数据中,员工的收入和学历呈中等程度的相关性,但是在不同的收入区间中,二者呈现不同于总体的相关性,比如在中等收入区间,学历和收入呈现更强的相关性,但是在更高的收入区间,学历与收入呈现接近相互独立的关系。借助准确的统计信息和有效的算法,能够发现数据分布区域中具有特殊相关性的子区域,对于

到稿日期:2014-09-23 返修日期:2014-12-09 本文受国家自然科学基金项目(61202331,61170013),软件工程国家重点实验室开放研究基金项目(SKLSE2012-09-33)资助。

曹 巍(1975—),女,博士,讲师,CCF 会员,主要研究方向为高性能数据库、数据库自管理自调优、闪存数据库等;王秋月(1974—),女,博士,讲师,CCF 会员,主要研究方向为数据库与信息系统、信息检索、知识库、自然语言问答等,E-mail:qiuyuew@ruc.edu.cn(通信作者);覃雄派(1971—),男,博士,讲师,CCF 会员,主要研究方向为高性能数据库、大数据分析等;王 珊(1944—),女,教授,博士生导师,CCF 高级会员,主要研究方向为高性能数据库、内存数据库、非结构化数据库、数据仓库与数据分析等。

关系数据库查询优化、数据分析等具有重要的价值，对决策支持可以提供更多的支持和深入的视角。

本文首次提出了数据相关性检测的新方法 HY-COCA (HYbrid distribution's COrrelation Coefficient based Association detection)，这种方法既可以计算出总体数据分布上的数据相关性程度，也可以发现隐藏在数据分布子区域中的特殊数据相关性。最后通过实验验证了 HY-COCA 方法的有效性。

2 相关工作

近年来，随着普适计算、数据中心和云计算环境的出现和兴起，计算机存储和处理的数据量规模呈几何级数增长。面对如此海量的数据，如何运用新方法发现数据中隐含的统计规律，对数据管理和数据分析以及传统关系数据库的查询优化都显得尤其重要。最近几年的数据管理顶级会议上，相当数量的工作围绕这一主题展开。

COCA[1]是一种能够精确度量属性间相关性的方法，它利用数据相关系数[9]，既可以在数据库的环境中给出最相关的若干属性对，又可以为给定的一对属性计算它们的相关程度。作为这种方法的基础，数据相关系数[9]是一对介于[0,1]的小数，分别表示两列之间双向数据依赖的程度。根据文献[1]的分析，这对相关系数有如下优点：(1)是双向依赖关系的度量；(2)实现简单；(3)指标同一性。

但是 COCA 对二维数据分布上子区域的数据相关性检测没有给出办法。

与 COCA 的工作最接近的是 CORDS[4]。CORDS 的原理是利用二维数据分布上的卡方检验检测属性对之间的关联程度，利用属性值个数等统计信息检测弱函数依赖。CORDS 方法对数据库中所有的属性对进行检验，输出关联程度由高到低的前 $K1$ 个属性对和弱函数依赖程度从高到低的前 $K2$ 个属性对。但这种方法是一种定性的序列方法，无法准确地检验给定一对属性的相关程度。另外，CORDS 依赖的卡方检验在统计学上要限制卡方分布的自由度，因此 CORDS 在采样数据上进行检测；为了解决卡方检验可信性的限制，列联表(即分布矩阵)至少应有 80%的格子频数大于 5[4]。CORDS 采用了压缩数据分布矩阵和利用系统目录中统计信息的方法，其中压缩数据分布会导致相关信息的丢失。CORDS 检测弱函数依赖使用的利用属性值个数的方法，增加了对弱函数依赖的漏报率(false negative)。

在给定二维数据分布上 CORDS 采用的是定性的假设检验方法，无法解决针对数据分布上的子区域定量检测其局部数据相关性的问题。

其他的相关工作还有 DB-Histograms[6] 和 SASH[7]，这两种方法针对高维数据空间，用统计图模型刻画属性组之间的相关性结构，目的是要在高维数据空间中找出若干个属性组，属性组内部的各属性之间具有较高的相关性，属性组之间是相互独立的，为这些属性组创建模拟联合分布的统计信息。二者的不同在于 DB-Histograms 是完全从数据出发，而 SASH 利用查询反馈或者更新查询优化的估计值，或者重新创建统计信息。但这两种方法都无法检测数据分布空间中子区域的特殊相关性。

在研究二元数据分布上子区域内部的数据相关性中，还有一个问题就是如何划分子区域。多维索引和多维直方图都提出了不同的多维空间的划分方法，这两种数据结构尽管应用目的不同，但在子区域划分的方法上多有重叠[8]。总体来说，对数据空间的划分有如下几种方法[8]。

Grid 划分方法，如多维索引中的 grid 文件[18]。STGrid[16]直方图、GenHist[15]直方图(GenHist 在不同的迭代间子区域可以重叠的特性类似于 R 树结构)等都采用此种划分子区域方法。这种方法将 D 个维度的每一维划分成 G_i ($1\leqslant i\leqslant D$)个区间，整个数据分布区域分成 $\prod G_i$ 个子区域。KDB 树[18]多维索引采用的划分方法与等深多维直方图的划分方法类似，按照事先规定的顺序依次划分各维空间。MHist 直方图的空间划分方法类似于 KDB 树多维索引，但空间划分的维度并不事先固定，而是选择最需要划分的维度进行划分。此外还有多维索引中的 quad 树[19]划分方法，每次的递归划分都将子区域划分为 NE、NW、SW、SE 4 个区域。

在这几种空间区域的划分方法(Grid 方法及其变种、KDB、MHist、quad)中，Grid 方法及其变种、KDB 方法和 quad 方法都是按照预先设定的划分模式划分区域，这样的方法能解决在数据空间中创建多维索引或者在数据分布矩阵中创建多维直方图的问题，但并不适合于在数据分布矩阵中搜索具有某一统计特征的子区域。MHist 方法[5]用有效的统计信息作为划分的依据，建立更准确的多维直方图，其采用的统计信息(比如面积、频率等[10])和统计方法(比如 MaxDiff、V-optimal[10]等)都旨在提高直方图准确率。

在我们的研究中，要解决的问题是搜索隐藏在数据分布矩阵中的具有特殊相关性的子区域，这是一种对划分的搜索，Grid 文件、KDB 树和 quad 树的方法都是按照事先规定的策略进行划分，并不适合对划分的搜索。数据相关系数能够准确地描述属性间的相关性，我们可以充分利用这一指标实现基于相关系数的动态子区域划分，在实现时采用类似 MHist 的方法搜索最合适的划分。

3 相关概念

数据集：在关系数据库中，对应一个基本表中的数据或者一次连接运算的中间结果。

一维数据分布：给定数据集 S 和属性 A，属性 A 上的取值及其在数据集中的出现频率所组成的二元组的集合，形式化为$\{(v_i,f_i)\mid v_i\in A$ 在 S 上的值域，f_i 为相应的取值在 S 上的出现频率$\}$。

二维数据分布：给定数据集 S 和属性对(A,B)，(A,B)的组合值及其在 S 中的出现频率所组成的二元组的集合，形式化为$\{((v_i,v_j),f_{ij})\mid v_i\in A$ 在 S 上的值域，$v_j\in B$ 在 S 上的值域，f_{ij} 为相应的组合值在 S 上的出现频率$\}$。

二维数据的相关性：给定数据集 S 和属性对(A,B)，根据数据集中的样本判断属性 A 和属性 B 之间的相互关联程度(或称依赖程度)。相关性的两种极端状况分别是函数依赖和相互独立。B 函数依赖于 A 是指在数据集 S 中，当属性 A 取值 a_i 时，属性 B 上取唯一值 b_j。A 与 B 相互独立是指在 S 中，当属性值 A 任取 a_{i1}，a_{i2}时，属性值 B 上的任意两个值 b_{j1}，b_{j2} 与之相对应的格子频率满足下式：

$f_{i1j1}/f_{i1j2}=f_{i2j1}/f_{i2j2}$，$\forall i1,i2\in[1,A$ 的值域大小$]$，$j1,j2\in[1,B$ 的值域大小$]$ (1)

判断属性 A 和 B 的相关性的方法主要是分析它们的联合分布矩阵，比如基于卡方检验的方法[4]和基于相关系数的检验方法 COCA[1]。

图 1 和图 2 是属性 A 和 B 的两个联合分布矩阵[2]。图 1 中，属性 A 和属性 B 具有函数依赖的关系。矩阵中 5 个 f_{ij} 代表 5 个非 0 的格子频率(未给出具体数值)，其余的格子频率为 0。图 2 是属性 A 和属性 B 完全相互独立的联合分布矩阵，矩阵中组合值的频率符合式(1)，其中 $c_i(i=2,\cdots,5)$可以是任意正有理数。

	b_1	b_2	b_3	b_4	b_5
a_1				f_{14}	
a_2		f_{22}			
a_3	f_{31}				
a_4			f_{43}		
a_5					f_{55}

图 1 函数依赖的二维数据分布矩阵

	b_1	b_2	b_3	b_4	b_5
a_1	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
a_2	c_2*f_{11}	c_2*f_{12}	c_2*f_{13}	c_2*f_{14}	c_2*f_{15}
a_3	c_3*f_{11}	c_3*f_{12}	c_3*f_{13}	c_3*f_{14}	c_3*f_{15}
a_4	c_4*f_{11}	c_4*f_{12}	c_4*f_{13}	c_4*f_{14}	c_4*f_{15}
a_5	c_5*f_{11}	c_5*f_{12}	c_5*f_{13}	c_5*f_{14}	c_5*f_{15}

图 2 完全相互独立的二维数据分布矩阵(值域密度最高，组合频率符合式(1))

但是，在现实世界中，数据间完全的函数依赖或者相互独立比较少见，常常是近似的函数依赖或近似相互独立。

二元数据分布的值域密度是一个大致描述数据分布中不同的单维属性值个数和多维组合属性值个数之间关系的指标。在数据集中，属性 A 有 D_A 个不同属性值，属性 B 有 D_B 个不同属性值，属性 A 与 B 的组合有 $D_{A,B}$ 个不同的组合值，则值域密度定义为 $\rho=D_{A,B}/(D_A*D_B)$。这是一个表示数据分布空间稠密程度的指标，但并不能准确描述属性 A 与 B 之间的相关性。

混合型多维数据分布：混合型数据分布有两种基本型[2]。

(1)总体上近似函数依赖的数据分布空间上，出现近似相互独立的局部区域，如图 3 所示。

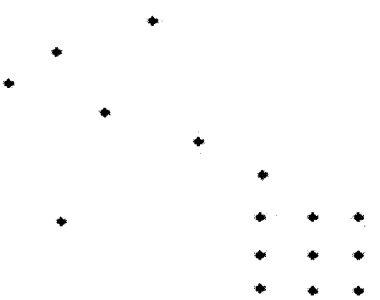

图 3 基本型(1)

(2)总体上近似相互独立的数据分布空间上，出现近似函数依赖的局部区域，如图 4 所示。

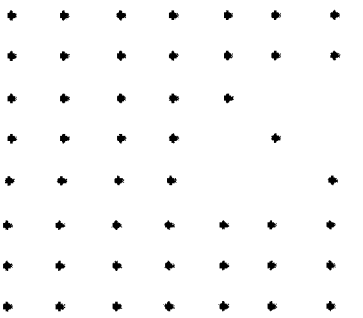

图 4 基本型(2)

混合型数据分布就是这两种基本型相互叠加、组合而形成的多维数据分布。本文将在数据分布中出现的局部近似相互独立和局部近似函数依赖的数据子区域称为特殊相关性子区域。HY-COCA 的目的就是在一般相关性的数据分布中，找到隐藏在其中的特殊相关性子区域。

基于样本的熵相关系数[9]的定义：

$$r_{j\to i}=-\frac{\sum_{i=1}^{d_1}\sum_{j=1}^{d_2}n_{ij}\ln(nn_{ij}/n_{i.}n_{.j})}{\sum_{i=1}^{d_1}n_{i.}\ln(n_{i.}/n)} \tag{2}$$

$$r_{i\to j}=-\frac{\sum_{i=1}^{d_1}\sum_{j=1}^{d_2}n_{ij}\ln(nn_{ij}/n_{i.}n_{.j})}{\sum_{j=1}^{d_2}n_{.j}\ln(n_{.j}/n)} \tag{3}$$

熵相关系数是一对在$[0,1]$之间的小数，它们是在二维数据分布矩阵上计算得到的。式中的符号解释如下：数据集 S 中属性 A 有 d_1 个不同的属性值，属性 B 有 d_2 个不同的属性值；n_{ij} 是组合属性值(a_i,b_j)的格子频率，$n_{i.}$ 是属性值 a_i 的边缘频率，$n_{.j}$是属性值 b_j 的边缘频率，n 为数据集 S 中的元组数，ln 是自然对数。两个相关系数分别表示属性 B 的取值以多大的概率能决定属性 A 的取值，或者相反，属性 A 的取值以多大的概率能决定属性 B 的取值。式(2)成立的条件是 $d_1\geqslant 2$；式(3)的成立条件是 $d_2\geqslant 2$。在后文中用 $CF_{A->B}$ 和 $CF_{B->A}$(Correlation co-eFficient，CF)简称这一对相关系数。

数据分布区域及其规模是 HY-COCA 中用到的概念。

数据分布区域：在数据分布矩阵中的一个矩形区域，用左上角和右下角的“坐标”来表示这个分布区域的范围，坐标实际是属性 A 和属性 B 在数据集上的取值，数据分布矩阵在 A 和 B 的维度上对应的取值都是按照从小到大的顺序排布。通常表示为$((a_k,b_l),\cdots,(a_m,b_n))$，其中 $m\geqslant k,n\geqslant l$ 是相应的属性值在各自维度上的序号。在实验阶段为了简化，也直接用序号作为数据分布区域的坐标，即$((k,l),\cdots,(m,n))$。

数据分布区域的规模 V：不能单纯以数据分布区域的范围大小作为数据分布区域的规模，因为$((a_k,b_l),\cdots,(a_m,b_n))$范围中有很多组合值并不在数据集 S 中存在，所以可以将$((a_k,b_l),\cdots,(a_m,b_n))$中存在的不同组合值个数作为数据分布区域的规模。$V=\#\{((v_i,v_j),f_{ij})\mid f_{ij}\neq 0\}$。

4 二维混合数据分布下相关性检测

4.1 问题描述和 Naive 算法分析

二维混合数据分布下相关性检测的问题可以描述为：给定关系数据库中的表(可以是基本表或者自然连接运算的结果)和用户感兴趣的属性对(A,B)，给出在总体数据分布上

(A,B)之间的相关程度，并找出其相关性明显不同于总体的子区域即具有特殊相关性的子区域。

为解决这一问题，一种 Naive 算法穷举所有的子区域，并判断每一个子区域的相关性是否足够特殊。为简化计算，假设数据分布矩阵为 $n*n$，则 Naive 算法需要处理的可能子区域为集合{(行数$=r\#$，列数$=c\#$)|$2\leqslant r\#\leqslant n$，$2\leqslant c\#\leqslant n$}。根据计算相关系数公式的限制，子区域必须满足 $2*2$ 以上的规模，则 $n*n$ 分布矩阵中，满足这个条件的子区域数目 $\|Q_n\|$ 为 $n^2(n-1)^2/4$。

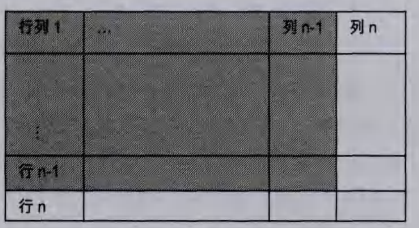

图 5　在数据分布矩阵中搜索子区域个数

证明：(归纳法)当 $n=2$ 时，$n*n$ 分布矩阵中只能有一个子区域，公式成立；当 $n>2$ 时，如图 5 所示，所有可能的子区域数等于阴影部分中包含的子区域数 $\|Q_{n-1}\|$ 加上所有至少包含列 n 或者行 n 的子区域 Δ 的个数 $\|\Delta\|$。根据归纳假设，$\|Q_{n-1}\|=(n-1)^2(n-2)^2/4$。下面分析 $\|\Delta\|$ 的取值。先只看 Δ 中包含列 n 的情况：子区域的行数可以有 $2,\cdots,n$，共有 $n-1$ 种不同的情况，用(#_of_row，#_of_possible_row_combination)表示，分别为$(2,n-1)$，$(3,n-2)$，$\cdots$，$(n,1)$，比如两行的区域可以为行 1 与行 2、行 2 与行 3、$\cdots$、行 $n-1$ 与行 n，共有 $n-1$ 种不同的情况，其他情况类推。每种行数的区域中，包含新增的列 n 的情况有 $n-1$ 种，包括列数为 2(列 $n-1$ 与列 n)、列数为 3(列 $n-2$，列 $n-1$，列 n)、$\cdots$、列数为 n(列 1，列 2，$\cdots$，列 n)，因此 Δ 中包含列 n 的个数 $\|\Delta_n\|$ 为

$(n-1)*[(n-1)+(n-2)+\cdots+1]=(n-1)*n*(n-1)/2$

同理 Δ 包含行 n 的个数 $\|\Delta_{rn}\|$ 也为$(n-1)*n*(n-1)/2$，因此 $\|\Delta\|=\|\Delta_{cn}\|+\|\Delta_{rn}\|-(n-1)^2$。其中第三项$(n-1)^2$ 为重复计算的既包含列 n 又包含行 n 的所有子区域个数。计算可得 $\|Q_{n-1}\|+\|\Delta\|=n^2(n-1)^2/4$。得证。

通过上述分析，可知 Naive 算法穷举子区域的复杂度为 $O(n^4)$，每一个子区域内计算相关系数的复杂度为 $O(n^2)$，因此 Naive 算法的复杂度是 $O(n^6)$ 规模的。

4.2　HY-COCA 算法描述

穷举法搜索所有的子区域并判断其相关性是否足够特殊，这种方法的时间复杂度非常高($O(n^6)$)。HY-COCA 算法采用启发式策略裁剪子区域的搜索空间。具体来说，按照相关性差异最大来划分子区域，从而找到具有特殊相关性的子区域。因为数据分布区域的相关系数在[0,1]之间，特殊相关性(包括完全函数依赖和完全相互独立)分别对应相关系数 1 和 0，如果一直按照相关性差异最大来划分子区域，当数据分布中隐藏有特殊相关性子区域时，这样的划分一定会逐渐逼近这些具有特殊相关性的子区域。启发式的方法可以有效地减小穷举法的搜索空间。

通过进一步细化为以下几个子问题，来尝试解决该问题。

(1)HY-COCA 的划分方法：解决的问题是，给定二维数据分布矩阵，如何进行划分以找到相关性差异较大的子区域。

(2)相关性差异的定义：解决的问题是，给定二维数据分布矩阵按照方法(1)进行划分时，如何定义子区域间的相关性差异较大，从而决定如何划分。

(3)HY-COCA 的结果返回：如果问题(1)、(2)得到解决，如何将得到的相关性差异较大的子区域返回给用户。

下面将分别从这 3 个方面介绍 HY-COCA 算法。

4.2.1　HY-COCA 的划分方法

在文献[5]中提到，MHist 多维直方图创建时以单维的统计信息为选择和划分子区域的依据。MaxDiff 划分方法在单维的边缘分布上，按照属性值排序，当相邻的统计信息之差大于某一阈值时则在该维的此位置上设置一个划分界限。MHist-2 每次设置一个界限，即将当前区域划分为两个子区域。

MHist 划分方法的特点是实现简单，因为其划分依据单维的统计信息，但是弊端就是依据单维的统计信息划分二维分布矩阵，划分的结果并不准确。特别是针对 4.1 节中的混合分布下相关性检验的问题，这种降维的划分方法不能保证准确地找到具有特殊相关性的分布子区域。因此我们依据二维的统计信息(相关系数)查看每一个可能的界限将当前区域划分成的两个子区域(假定一个可能的划分将一个给定区域划分成两个非单行非单列的区域)，只有两个子区域的相关性差异达到最大时，才实施相应的划分。

具体来说，HY-COCA 按照相关性差异划分子区域的算法类似 MHist-2 的递归方法(但划分的依据是二维的统计信息)，可以简要描述为：

(1)检查集合 P 中的每一个区域，为其计算每一个可能划分的相关性差异分数，所有可能的划分中最大的差异分数设置为该区域的相关性差异分数(简称为区域分数)；

(2)找到 P 中具有最大区域分数的区域 R_{corr}^{max}，将其按照分数计算的划分方法进行划分，划分出的两个新区域放回 P 中，并将 R_{corr}^{max} 从 P 中删除，回到步骤(1)。

这一问题还需考虑算法何时结束。MHist-2 的划分算法的结束条件是当 P 中的区域个数达到某一个数目。HY-COCA 算法的目的是找到具有显著相关性差异的子区域，而这样的子区域数目事先是不确定的，因此可以考虑当划分出来的子区域相关性差异不明显时结束循环。但是如果数据的值域空间很大，而隐藏的特殊相关性子区域较小，则很可能还没有找到这些特殊相关性区域，算法就因为没有显著的相关性差异而终止划分搜索。

因此考虑使划分搜索结束的两个条件：一个是关于相关性的最大区域分数的限制条件，另一个是数据分布区域的规模因素。如果相关性的最大区域分数不符合限制条件，则判断 P 中最大的分布区域是否大于规模阈值，如果是，则找到 P 中规模最大的分布区域，将其沿较大的维度等分成两个新的子区域，代替 P 中原来的区域，继续搜索划分。只有当相关性差异的最大区域分数不符合限制条件并且数据分布区域的规模小于阈值时，算法终止。

针对数据分布区域的规模，HY-COCA 采用了事先预设阈值的方法，但是针对相关性差异的划分，基本型(1)和基本型(2)的混合数据分布的相关性差异分数的取值是不一样的，无法用统一的事先设定的阈值来检测。因此 HY-COCA 采用与父节点的区域分数进行比较的方法，将最大区域分数的

限制条件设为：若 R_{corr}^{max} 的区域分数表示为 $SCORE_{corr}^{R\_CORR\_MAX}$，其父节点的区域分数为 $SCORE_{corr}^{F(R\_CORR\_MAX)}$，如果 $SCORE_{corr}^{R\_CORR\_MAX} > SCORE_{corr}^{F(R\_CORR\_MAX)} * factor$，则意味着相关性差异足够显著。其中 *factor* 随各次递归以 $(1-a)^i$ 的速率变化，$i \geqslant 0$ 代表各次按相关性最大差异进行划分的迭代次数。该条件限制 $SCORE_{corr}^{R\_CORR\_MAX}$ 的下限值，*factor* 大于 0 且小于 1，允许当前区域的相关性差异分数小于父区域的相关性差异分数，但必须大于某个比例，*factor* 逐渐递减，意味着 HY-COCA 希望能发现更多具有相关性差异的数据分布区域；*factor* 同时也是一个收敛因子，*factor* 收敛得越快（a 越大，a 称为收敛参数），HY-COCA 收敛得越慢，结果中会有很多微小的不显著的特殊分布区域。经过实验，发现当 $a=0.2$ 时，算法的收敛和针对不同类型的数据分布其相关性检验准确度均比较优化，具体实验结果参见 5.3 小节。

因此，HY-COCA 为了找到特殊相关性的子区域，结合了相关性差异最大和划分最大规模分布区域两种划分方式。

初始情况下，*P* 中只有一个区域，即整个数据分布区域。图 6 为算法的简单描述。

算法：fn_max_coca_score_region
输入：P
输出：P 中具有最大差异分数的区域和其差异分数
算法描述：（遍历 P 中每一个区域，需要时为其计算差异分数，返回最大差异分数及其区域）
For P 中每一个区域 R
　If R 的差异分数 $SCORE_{corr}^{R}$ 为空 then 调用 coca_score(R)计算 R 的差异分数 endif
End For
Return 具有最大 $SCORE_{corr}^{R}$ 的区域 R_{corr}^{max} 和最大的差异分数 $SCORE_{corr}^{R\_CORR\_MAX}$。

算法：HY-COCA
输入：P 的初始状态，数据分布区域的规模阈值 T_{size}，a
输出：P 中包含按照相关性最大差异和按规模划分出来的分布区域的集合（划分树）
算法描述：
//P 中最大差异分数及其区域
R_{corr}^{max}，$SCORE_{corr}^{R\_CORR\_MAX}$←fn_max_coca_score_region(P)
//初始 P 中只有一个分布区域；父亲结点 F(R_CORR_MAX)为空
F(R_CORR_MAX)←NULL；$SCORE_{corr}^{F(R\_CORR\_MAX)}=0$
factor←1
while $SCORE_{corr}^{R\_CORR\_MAX} > SCORE_{corr}^{F(R\_CORR\_MAX)} * factor$ OR P 中最大区域尺寸 > T_{size} do
　//判断是否相关性差异划分
　if $SCORE_{corr}^{R\_CORR\_MAX} > SCORE_{corr}^{F(R\_CORR\_MAX)} * factor$
　　split R_{corr}^{max} along the biggest correlation differentiating dividing, getting R_{cld1}，R_{cld2}；
　　factor←factor * (1−a)
　else
　　//否则，则取规模最大区域划分
　　R_{size}^{max}←P 中规模最大的区域；
　　$Side_{max}$←R_{size}^{max} 区域中较长的边；
　　Split R_{size}^{max} at the middle point of $Side_{max}$, getting R_{cld1}, R_{cld2}；
　End if
　Add R_{cld1}，R_{cld2} to P；deleting R_{corr}^{max} from P
　R_{corr}^{max}，$SCORE_{corr}^{R\_CORR\_MAX}$←fn_max_coca_score_region(P)
　//P 中最大差异分数及其区域
　F(R_CORR_MAX)←R_{corr}^{max}的父节点
　//返回其父节点
End while

图 6　HY-COCA 算法描述

给定一个数据分布区域 $R((a_k,b_l),\cdots,(a_m,b_n))$，用函数 *coca_score*(*R*)计算其相关性差异分数。*R* 在 *A* 和 *B* 维度上的投影分别为 $[a_k,a_m]$ 和 $[b_l,b_n]$，其中 $m>k$ 并且 $n>l$；依据相关系数公式成立的限定条件，这个区域必须至少有一个维度的长度大于 2，即 $m>k+2$ 或者 $n>l+2$，才是一个可划分的数据分布区域。考察相关性差异的划分时，如果 $m>k+2$，则有 $m-k-2$ 种可能的划分，划分出来的两个子区域在 *A* 和 *B* 维上的投影为 $[a_k,a_i]$ 和 $[b_l,b_n]$、$[a_{i+1},a_m]$ 和 $[b_l,b_n]$，其中 $k+1 \leqslant i \leqslant m-2$；如果 $n>l+2$，则有 $n-l-2$ 种可能的情况，划分出来的两个子区域在 *A* 和 *B* 维上的投影为 $[a_k,a_m]$ 和 $[b_l,b_j]$、$[a_k,a_m]$ 和 $[b_{j+1},b_n]$，其中 $l+1 \leqslant j \leqslant n-2$。HY-COCA 计算两个维度上所有可能划分出来的成对子区域的相关系数及其相关性差异，找出其中相关性差异最大的划分和分数作为 *R* 的相关性差异划分和相关性差异分数。

复杂度分析：时间复杂度上，HY-COCA 划分算法的复杂度（无论是按相关性差异最大划分，还是按照规模划分）为 $O(n)$ 级别。假设 *T* 为 HY-COCA 划分算法的时间复杂度，并且每次将整个数据区域划分成两个相等的子区域，可以递归表示为 $T(n)=T(\lfloor n/2 \rfloor)+T(\lceil n/2 \rceil)$，则可以证明 $T(n)=O(n)$[14]。

而 *coca_score*(*R*) 的时间复杂度是 $O(n^2)$ 级别的，因此 HY-COCA 的算法时间复杂度是 $O(n^3)$ 级别。为减少重复计算量，*coca_score*(*R*) 采用两个局部矩阵辅助存储两个维度上所有可能划分的边缘分布，得到子区域的差异分数。因此，HY-COCA 的空间复杂度是 $\Theta(n^2)$ 级别，*n* 代表的是数据分布矩阵的规模，并不是数据量的大小。

4.2.2　相关性差异的定义

前述方法的一个关键点就是如何判断潜在的两个子区域相关性差异最大。文献[2-4]中都采用了值域密度的指标，值域密度在涉及多维直方图的创建[2,3]或者判断弱函数依赖中的个别情况[4]时有意义，但是在解决 HY-COCA 面临的问题时意义不大。如图 7 和图 8 所示，两个分布矩阵均可判断为近似函数依赖，但是两个分布矩阵的值域密度却是一个最小，一个最大。

如果以值域密度来评判相关性的差别，很可能会忽略如图 8 的分布情况，CORDS 用这种方法则会造成漏判率（false negative）的升高。相关系数能够很准确地刻画数据间的相关程度，因此，在 HY-COCA 中采用相关系数评价子区域相关性的差别。

	b_1	b_2	b_3	b_4	b_5
a_1		123			
a_2			227		
a_3				229	
a_4	122				
a_5					330

图 7　值域密度最小（20%）的情况下，函数依赖的数据分布（此时两个相关系数均为 1）

	b_1	b_2	b_3	b_4	b_5
a_1	1	123	1	1	1
a_2	1	1	227	1	1
a_3	1	1	1	229	1
a_4	122	1	1	1	1
a_5	1	1	1	1	330

图 8　值域密度最大(100%)的情况下,近似函数依赖的数据分布(此时两个相关系数均为 0.92262)

那么另外一个问题就是,如何判断子区域之间相关性的差别足够大。具体地说,也就是用什么样的分数表示可能划分的两个子区域的相关性及其差异。

由于相关系数的取值范围是[0,1],第一种方法可以直接用一对相关系数的最大值表示子区域的相关性分数,用两个子区域的相关性分数之差的绝对值作为差异分数。若当前区域为 R,考察它的一个可能的垂直划分,将 R 分成 R_L 和 R_R 两个子区域,两个子区域分别具有两组成对的相关系数,分别为 $CF^L_{A->B}$,$CF^L_{B->A}$,$CF^R_{A->B}$,$CF^R_{B->A}$,则这个划分的差异分数形式化表示为 $|MAX(CF^L_{A->B},CF^L_{B->A})-MAX(CF^R_{A->B},CF^R_{B->A})|$。

但在实验中发现,单纯的相关系数的大小会受到区域不同维度属性值个数(如 $|D_A|$ 与 $|D_B|$,$|D_A|^L$ 与 $|D_B|^L$,或者 $|D_A|^R$ 与 $|D_B|^R$)的影响。当前分布区域的两个维度上不同值个数相差很大时,两个相关系数差别也很大,比如当 $|D_B| \ll |D_A|$ 时,$CF_{A->B} \gg CF_{B->A}$,很可能出现的一种情况就是,大多数区域的划分都是沿划分维度的前两行(或前两列)进行——因为这是能划分出的子区域的最小行数(或列数),这样就造成了区域划分的不公平。

为避免这种情况,在计算相关性分数时需要将不同属性值个数的因素消除,因此 HY-COCA 采用的相关性分数计算逻辑如图 9 所示。

输入:一个分布区域的相关系数 $CF_{A->B}$ 和 $CF_{B->A}$,该区域的行数 D_A 和列数 D_B(即单维上的不同值个数)

输出:调整后的该区域的相关性分数 $SCORE_{corr}$

算法功能:消除不同属性值个数对相关系数的影响,调整原有相关性分数使其更能反映一般意义上的相关性。

算法 computeCorrScore 描述:

If $CF_{A->B} \geqslant CF_{B->A}$ AND $D_B \leqslant D_A$ THEN

$SCORE_{corr} = CF_{A->B} * D_B/D_A + CF_{B->A} * (1-D_B/D_A)$

Else if $CF_{A->B} \leqslant CF_{B->A}$ AND $D_B \geqslant D_A$ THEN

$SCORE_{corr} = CF_{B->A} * D_A/D_B + CF_{A->B} * (1-D_A/D_B)$

Else

$SCORE_{corr} = MAX(CF_{A->B}, CF_{B->A})$

Endif

图 9　相关性分数计算方法

经过这样的调整,区域的相关性分数的取值仍然在[0,1]之间。同时考虑了这样的情况,如果某一方向的相关系数(比如 $CF_{A->B}$)由于相应的属性值个数相对少(如 $D_B \leqslant D_A$)而高于另一相关系数,则将该相关系数乘以一个缩小因子(如 D_B/D_A)作为权数参与计算相关性分数,由于两个相关系数的权数之和为 1,因此相关性分数的取值范围不发生改变。

同样地,前述划分的差异分数可以用两个子区域的相关性分数之差的绝对值来表示,即 $|SCORE^L_{corr}-SCORE^R_{corr}|$。

4.2.3　HY-COCA 的结果返回

解决前述两个问题后,最后一个问题就是,将什么样的结果返回给用户,如何返回这样的结果。

如 4.2.1 节所述,对数据分布区域的划分有按相关性差异最大和划分最大规模两种划分方式,HY-COCA 面临的问题是要找出具有特殊相关性的区域,那些因为没有显著的相关性差异而选择按划分最大规模的区域,只是为进一步找出特殊相关性子区域而进行的权宜划分,因此不需要返回给用户。

对于待返回的特殊相关性子区域,HY-COCA 可以使用用户事先设定(或者默认)的近似函数依赖和近似相互独立的相关系数阈值来确定,同时还可以让用户设定(或 HY-COCA 默认)数据分布区域的规模阈值,返回足够规模的具有特殊相关性的分布区域。

直观说来,HY-COCA 为检测特殊相关性子区域,对数据分布矩阵的划分,可以表示为一棵从根向叶生长的二叉树。在 HY-COCA 中,将这样的一棵树叫做划分树。树中每一个节点对应了一次划分产生的数据分布区域。根是整个数据分布矩阵,叶子节点对应集合 P。每一次划分都将当前区域划分成两个子区域,划分分为两种类型:一类是按照相关性差异最大划分,一类是划分最大规模的子区域,每一次划分都产生新的叶节点,需要重新调整 P 中的元素。HY-COCA 算法的一个中间状态可用图 10 描述。图中填充加点的矩形代表候选结果,是需要进一步判断的具有相关性差异的子区域。

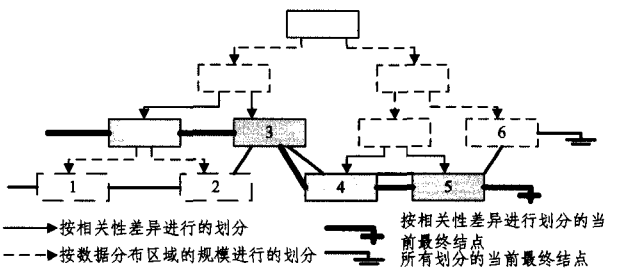

图 10　HY-COCA 划分算法的演进图示——划分树

图 10 中,实线箭头及其末端的实线矩形表示按照相关性差异最大进行的划分及其子区域(相关性差异节点),虚线的部分表示划分最大规模的分布区域及其等分的子区域(规模划分节点)。加粗实线连接的是划分树的叶节点,即 P 中的元素。如果此时算法结束需要返回给用户具有特殊相关性的子区域,P 中 6 个子区域,只有 3 个实线框表示的矩形(即矩形 3、4、5)可能是特殊相关性的子区域。对于 P 中其他按照规模划分出来的子区域,HY-COCA 采用的策略是:如果这样的子区域单个出现(如矩形 6),则放弃之;如果这样的子区域成对出现,左右兄弟同时出现在 P 中(如矩形 1 和 2),则回溯至其父节点;若其父亲节点也是按照规模划分出来的子区域,则按照前述原则递归处理,直到回溯至其按照相关性差异最大划分出来的第一个祖先节点(图中回至矩形 1′),然后,用相关性阈值和规模阈值判断这些按相关性差异最大划分出来的子区域是否具有显著特殊的相关性并且规模足够显著,作为检测结果返回,算法见图 11。

实验中发现,在算法 pruneSizePartition 返回的子区域集合 P' 中,可能会有相邻的子区域,比如两个子区域 $R1$ 和 $R2$ 的简化坐标分别为((152,128),…,(167…139))和((106,128),…,(151…139)),当这两个子区域的 $CF2$ 均接近于 1

时，或者其 $CF1$ 均接近于 0 时，可以认为这两个子区域的数据分布模式一致，将其沿属性 A 维度合并为一个结果子区域（(106,128),…,(167,139)）。沿属性 B 维度的合并同理。经过合并后的检测结果集合称为 P'_{merged}，其中那些符合最终输出条件的区域集合是用户最终能看到的，记为 P'_{final}。

算法：pruneSizePartitions

输入：划分树 T，划分树叶节点的链表 P

输出：T 中按相关性差异划分的最接近叶子节点的区域集合 P'（包括 P 中原有的按相关性差异划分的子区域，P 中按规模划分的区域的最近的相关性差异祖先区域）

描述：

```
While(P 中有规模划分节点)
    For P 中每一个规模划分节点 N
        If N 是其父亲 F 的左儿子，Then 将 N 从 P 中删除，F 左儿子设为空；
        Else if N 是其父亲 F 的右儿子，且 F 的左儿子节点为空，then 将 N 从 P 中删除，F 右儿子设为空；将 F 加入 P 中；
        Else 将 N 从 P 中删除，F 右儿子设为空。
    NEXT FOR
NEXT WHILE
返回新的集合 P'
```

图 11 pruneSizePartition 算法伪码（该算法将按照规模进行划分的最终结点剔除掉，留下按照相关性差异最大进行划分的最终结点）

5 实验

本节介绍 HY-COCA 的实验结果。首先用实验验证收敛参数 a 的取值方法，其次设计了两类实验来验证 HY-COCA 的有效性和性能，一类是生成数据的混合分布矩阵，一类是用基准测试的数据测试，两类实验结果显示 HY-COCA 能够在可接受的时间内检测出具有特殊分布的子区域。

实验的环境是 AMD Turion 64 X2 双核 CPU，主频 2GHz，内存 2GB。运行的操作系统是 Microsoft Windows Vista，编译环境是 Microsoft Visual C++ 2008 Express 版。

5.1 数据集

5.1.1 生成数据

文献[3]用的是生成不同相关性的数据分布的方法，大致可分成 3 种情况：a）函数依赖或近似函数依赖；b）完全相互独立或近似相互独立；c）随机生成的数据分布。

HY-COCA 增加了生成图 3 和图 4 的混合相关性的两种类型数据分布。第一种是在近似函数依赖的数据分布上生成若干个近似相互独立的数据分布子区域，简称为 IOF（Independence Over Functional dependence）；第二种是在近似相互独立的数据分布中生成若干个近似函数依赖的子区域，简称为 FOI（Functional dependence Over Independence）。

在实验中，指定在整个数据分布区域上生成 $\#special\_regions$ 个特殊相关性的数据分布区域。比如对于 IOF 类型的混合分布，先在给定的 $|A|*|B|$ 大小的矩阵上生成近似函数分布的数据分布，然后给定 $\#special\_regions$ 个不相交的子区域，在这些子区域中生成近似相互独立的数据分布。同样，对于 FOI 类型的混合分布，先在整个 $|A|*|B|$ 大小的矩阵上生成近似相互独立的数据分布，然后在给定的 $\#special\_regions$ 个不相交的子区域中生成近似函数依赖的数据分布。

HY-COCA 算法不仅能检测混合数据分布，对于非混合型分布的函数依赖或完全相互独立的数据分布，也应该能检测。因此对于几种非混合的数据分布类型，也用 HY-COCA 进行了检验，检验结果见 5.3 节。

5.1.2 基准测试 TPCH 的数据

选择 TPCH 的原因是 TPCH 测试主要针对 OLAP 的一些分析应用，对其数据中混合分布的相关性分析及其他的 OLAP 分析有辅助作用，同时 TPCH 的数据的本质更新不频繁，比较适合于 HY-COCA 的混合分布相关性分析，结果会较有说明性。

处理 TPCH 中的 DBGEN 程序生成的原始数据时，HY-COCA 先为原始数据计算数据分布矩阵，然后在数据分布矩阵之上进行特殊相关性区域的判断。在测试时，控制 DBGEN 生成的数据规模 $SF=0.01$。这里没有采用更高的 SF，是因为对于封闭值域的属性（属性值个数固定的属性，比如性别、部门等），数据分布矩阵的规模不会随着数据量的增大而改变，数据的相关性特征变化也不大；而经测试，对于开放值域的属性（属性值的个数随着数据量的增加而增加，比如值域为实数的属性、日期属性等），虽然数据分布矩阵的规模会随着数据量的增大而变大，但是属性间的相关性特征仍然保持，因此采用合适的数据规模即可检测其数据相关性特征。另一方面，因为 HY-COCA 的算法时间复杂度是 $O(n^3)$ 级别的，对于开放值域的属性，如果数据量很大，数据分布矩阵的规模使 HY-COCA 算法很难在可接受的时间内完成，这也促使我们对于大规模的数据，采用基于采样的混合数据分布关联性检测方法[1]，或者采用近似算法检测具有特殊相关性的子区域，这一部分工作我们会在未来完成。

5.2 HY-COCA 算法有效性的评价

使用生成的数据评价 HY-COCA 算法的有效性。生成具有特殊相关性的子区域个数为 $\#special\_regions$，每一个子区域的面积为 $Size_i$，$i\in[1,\#special\_regions]$。HY-COCA 报告检测出的具有特殊性相关性的子区域个数为 $\#detect\_regions$，每一个检测出的子区域与生成的 $\#special\_regions$ 个子区域的相交面积为 $overlap\_region\_size_j$，$j\in[1,\#detect\_regions]$。定义检出率和覆盖率如下所示：

$$det\_rate=\frac{\sum_{j=1}^{\#detect\_regions} overlap\_region\_size_j}{\sum_{i=1}^{\#special\_regions} size_i}$$

$$cover\_rate=\frac{\sum_{j=1}^{\#detect\_regions} overlap\_region\_size_j}{\sum_{i=1}^{\#detect\_regions} detect\_region\_size_i}$$

二者都是[0,1]之间的小数。det_rate 类似信息检索中的查全率（recall），它评价 HY-COCA 检测特殊数据相关性区域的完备程度，当 det_rate 为 1 时，意味 HY-COCA 准确地检出了所有特殊相关性的子区域，det_rate 越小，HY-COCA 检测特殊数据相关性子区域的完备程度越低；覆盖率类似信息检索中的查准率（precision），它评价 HY-COCA 检测的结果中真正具有特殊相关性的子区域的准确程度，$cover\_rate$ 越大意味 HY-COCA 的 true positive 越显著，det_rate 越小意味 HY-COCA 的 false negative 越显著。

5.3 实验结果及其分析

5.3.1 收敛参数 a 的选取

HY-COCA 检测的是未知类型的数据分布，收敛参数 a

的取值既影响对可能的混合分布的相关性检测的准确性，也影响对其他数据分布类型的相关性检验的效果。基于这一原则，在设计实验选择 a 取值时，分别针对 IOF 和 FOI 类型的混合数据分布，以及其他类型的分布（包括近似函数依赖、近似数据独立、随机数据分布）进行了实验。同时考虑 a 在不同范围内的典型取值 0.05，0.1，0.2，0.3，0.5，本实验未考虑大于 0.5 的情况，因为这时 $factor$（取值为 $(1-a)^i$，见 4.2.1 节）对候选子区域的选择率不高。

1）针对混合数据分布的结果

实验生成了几组不同规模的 FOI 和 IOF 型混合数据分布，得到了类似的趋势。现以 100 * 100 规模的数据分布为例，说明针对混合数据分布的实验结果。

生成数据的方法是在 100 * 100 规模的数据分布矩阵上生成函数依赖（或者数据相互独立）的数据分布，然后在随机的位置生成 20 个干扰数据，使其数据分布为近似函数依赖和近似数据相互独立，最后在随机的 8 个子区域生成数据相互独立（或者函数依赖）的数据分布，使总体为 FOI 类型（或 IOF 类型）的混合数据分布。

图 12 显示不同的 a 值对 HY-COCA 在 FOI 混合分布上的准确率没有影响，但是较小的 a 值（比如 0.05）会影响 HY-COCA 在 IOF 混合分布上的准确率，a 增大至 0.2 后，基本不会影响准确率。

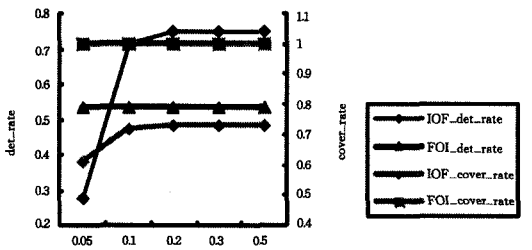

图 12　不同的 a 值对检出率和覆盖率的影响

2）针对其他非混合型数据分布的结果

a 的选取也应能支持对其他类型数据分布的相关性检测，这时 HY-COCA 检验出来特殊相关性的子区域实为干扰信息。同样在 100 * 100 的数据分布矩阵规模上生成了近似函数依赖、近似相互独立和随机数据分布，生成随机数据分布的方法是逐行生成 100 * 100 的数据分布矩阵，第 i 行生成随机多个（$Rand(i)$）非空频率组合值，这 $Rand(i)$ 个非空频率组合值随机出现在同一行 $Rand(i)$ 个不同的位置上。

实验结果显示，针对随机数据分布矩阵，P'_{merged} 中特殊子区域规模非常小，这些规模小的特殊相关性的子区域的实际意义并不大，可以通过设置最终结果子区域的规模阈值，减少 HY-COCA 报告干扰结果。近似函数依赖和近似相互独立的数据分布中情况类似，P'_{final} 并未报告出具有足够规模的特殊相关性的子区域。实验采用的检测标准暂定为集合 P'_{merged} 中的子区域个数（记为 det_count），det_count 越小，说明 HY-COCA 所发现的干扰因素越少。

图 13 示出针对 3 种类型的数据分布，规范化后的 det_count 个数在不同的 a 取值下的取值变化。规范化的方法是用当前的 det_count 值除以同组中最大的 det_count 值。

从图 13 可以看出，较小的 a 值有利于 HY-COCA 分析近似函数依赖的数据分布的相关性，而 a 值为 0.5 时有利于近似相互独立的数据分布相关性检测。综合 1 和 2 的实验结果，采用比较稳定的中间值 0.2 作为收敛参数 a 的取值。

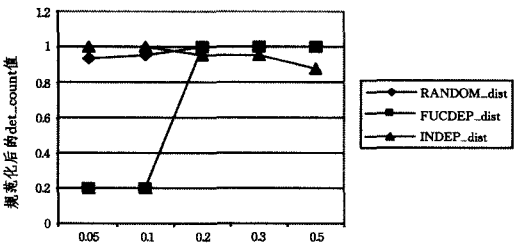

图 13　不同的 a 值对其他非混合型数据分布的影响

5.3.2　生成数据的实验结果

实验生成了很多种不同规格的混合型数据分布矩阵，来专门验证 HY-COCA 算法的有效性，现报告具有代表性的数据分布矩阵规模为 DM1：300 * 300、DM2：500 * 500、DM3：1000 * 1000、DM4：1500 * 1500，DM：Distribution Matrix 情况下 HY-COCA 的运行结果。对于每一种规模，都生成了 IOF 和 FOI 两种类型的混合分布，其中 DM1 生成 3 个特殊数据分布的子区域，如 DM1 的 IOF 数据分布，先在整个数据空间上生成近似函数依赖的数据分布，然后随机找到 3 个不相交的子区域，在这 3 个子区域中生成近似数据相互独立的数据分布；DM1 的 FOI 的混合数据分布，先在整个数据空间上生成近似数据相互独立的数据分布，然后在随机找到的 3 个不相交子区域中生成近似函数依赖的数据分布。实验结果中不同的数据分布标识为 DMX-y。数字 X 代表不同数据分布矩阵的编号，y 代表生成的具有特殊数据相关性的子区域的个数。

1）IOF 的数据分布的结果

图例中检出率的第一个参数，若为 merge，则 HY-COCA 为返回的特殊相关性区域采用了合并操作（见 4.2 节问题 3）；若为 no_merge，则没有采用子区域合并的方法。第二个参数代表最终返回的特殊相关性子区域的规模阈值（HY-COCA 采用区域中非空组合值的个数表示区域的规模），如果为 0 意味不限定返回子区域的规模，若为 10 则代表要求子区域中非空组合分布值的个数多于或等于 10。

图 14 显示，对于小规模的数据分布，并且特殊相关性的子区域不多的情况下，检出率接近于 1，DM4 的几种情况检出率稍低，均为 0.785。

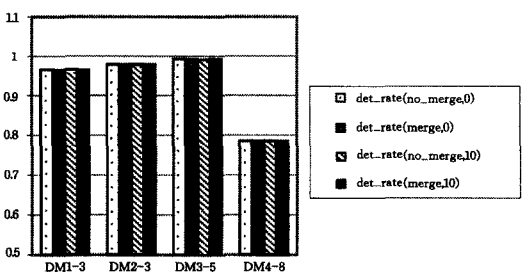

图 14　IOF 几种数据分布的检出率

从图 15 中可以看出，几种数据分布上的覆盖率不如检出率高，说明高检出率牺牲了一些准确率。通过比较可知，HY-COCA 检索出来的有特殊相关性的子区域的面积比真正具有特殊相关性的子区域面积大 29%～38%。另外 4.2.3 节中的合并算法将分布矩阵 DM3-5 的检出率提高 0.002，但也使其准确性降低 0.03。合并算法对 IOF 类型的数据分布的检出率的提高效果均不明显。

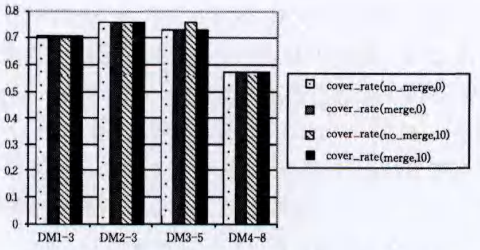

图 15　IOF 几种数据分布的覆盖率

2)FOI 的数据分布的结果

从图 16 中可以看出几种不同规模的 FOI 数据分布中，HY-COCA 检出率大多在 80%以上；在限制输出子区域的规模(子区域中非空组合分布值的个数大于或等于 10)时，合并算法在几种不同规模的 FOI 数据分布下，对 HY-COCA 的检出率均有提高，最明显的情况使检出率提高 9.6%(DM1-3 数据集)。

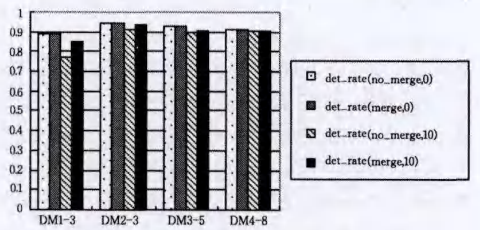

图 16　FOI 几种数据分布的检出率

图 17 中的 FOI 数据分布中，覆盖率均接近于 1，可以推断出 HY-COCA 算法找到的特殊相关性子区域是真正的特殊相关性子区域的真子集。此时，合并算法对进一步提高覆盖率意义不大。

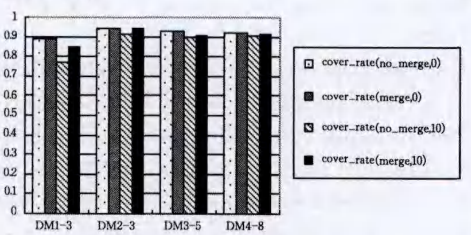

图 17　FOI 几种数据分布的覆盖率

3)时间效率评价

从图 18 中的时间效率来看，不同规模的数据分布矩阵上 HY-COCA 算法执行时间是呈不同的数量级的。HY-COCA 算法是 $O(n^3)$级别的，使得当数据分布矩阵的规模在 1500 * 1500(DM4-8)规模时，HY-COCA 在 IOF 和 FOI 两种数据分布上的时间效率比较低。

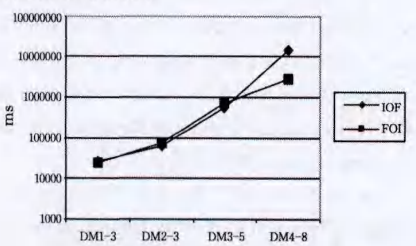

图 18　HY-COCA 在几种不同类型的数据分布上运行的时间

4)不同数目的特殊相关性子区域实验结果

在这部分实验中，固定数据分布矩阵为 DM3，规模 1000 * 1000，针对 IOF 型和 FOI 型的数据分布，在 DM3 上生成不同数目的特殊相关性子区域，如表 1 和表 2 所列。不同的子区域数目对 HY-COCA 的检出率、覆盖率和执行时间的影响分别见图 19 和图 20。

表 1　IOF 类型的不同数据分布特征

特殊相关性子区域的数目	非空组合值的个数	CF1	CF2
20	85045	0.267067	0.254248
50	85488	0.241741	0.235334
100	85547	0.241925	0.235484
150	89100	0.237264	0.249742

表 2　FOI 类型的不同数据分布特征

特殊相关性子区域的数目	非空组合值的个数	CF1	CF2
20	918375	0.011676	0.011680
50	918315	0.011668	0.011676
100	917901	0.011900	0.011907
150	904553	0.013356	0.013377

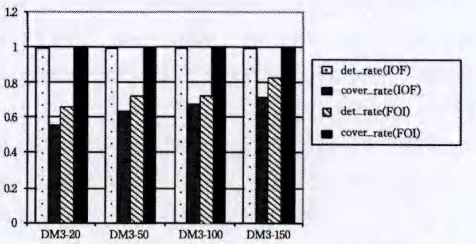

图 19　HY-COCA 在不同数目特殊相关性子区域情况下的有效性

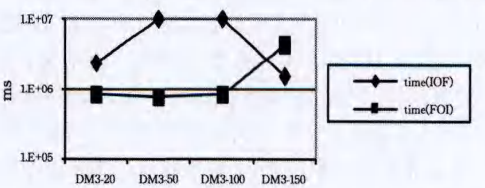

图 20　HY-COCA 在不同数目特殊相关性子区域情况下的效率

表 1 和表 2 中的数据分布特征，包括不同的组合值个数和相关系数等，都无法进一步揭示数据分布矩阵中隐藏的特殊相关性的数据分布子区域。图 19 显示，HY-COCA 检测出的具有特殊相关性的子区域，在 IOF 类型的数据分布矩阵中的检出率比较好，接近于 1，而覆盖率相对较低。在 FOI 类型的数据分布矩阵中，检出率相对较低，覆盖率却接近于 1。HY-COCA 的检出率与覆盖率就像信息检索中的查全率和查准率一样，是一对矛盾，这点在实验结果中也比较明显。从图 20 显示的时间效率上看，HY-COCA 工作在 FOI 类型的数据分布矩阵的时间效率总体上要高于 IOF。但对于这个规模的数据分布矩阵(DM3：1000 * 1000)，HY-COCA 的时间效率在千秒—万秒的数量级。

5.3.3　TPC-H 基准测试数据的实验结果

手工生成的方法可以控制数据分布，包括数据分布矩阵规模、数据分布类型、具有特殊相关性的子区域等，因此可以验证 HY-COCA 对这些预先设定的特殊相关性子区域的检验准确性。对于基准测试的数据，其相关性特征是事先不知道的。在用生成的数据验证了 HY-COCA 算法的有效性后，可以看到 HY-COCA 算法用在基准测试数据的检测结果，如

本节所示。为验证 HY-COCA 对 TPC-H 基准数据的检测，改进了 HY-COCA 算法，使其能够计算输入的原始数据的数据分布矩阵，并且除了支持整数类型的属性之外，还支持浮点数、字符串类型。

共测试了 5 对属性：Part 表的 P_MFGR，P_BRAND；Part 表的 P_CONTAINER，P_RETAILPRICE；LINEITEM 表的 L_QUANTITY，L_DISCOUNT；LINEITEM 表的 L_SUPPKEY，L_QUANTITY；PARTSUPP 表的 PS_SUPPKEY，PS_AVAILQTY。其中 PARTSUPP 表的 PS_SUPPKEY 和 PS_AVAILQTY 的数据分布矩阵规模是 1000 * 9996，一对相关系数是(0.690424，0.521451)，HY-COCA 花费了 24 个小时也没有算出结果，遂放弃，打算在未来工作中用采样或者近似方法解决这类大规模数据分布矩阵的相关性检验问题。其余几对属性的数据分布矩阵特征如表 3 所列。

表 3　TPC-H 的几对属性的数据分布矩阵特征

Attribute Pairs	属性 1 不同值个数	属性 2 不同值个数	不同组合值个数	相关系数	执行时间(ms)
API：P_MFGR，P_BRAND	5	25	25	(1，0.5)	78
APII：P_CONTAINER，P_RETAILPRICE	40	2899	18278	(0.475252，0.223382)	1372262
APIII：L_QUANTITY，L_DISCOUNT	50	11	550	(0.000109，0.000178)	1405
APIV：L_SUPPKEY，L_QUANTITY	1000	50	50000	(0.006056，0.010693)	48766

HY-COCA 检测的结果中，API 的数据分布是纯函数依赖，HY-COCA 的算法无需进一步检查子区域；APIII 和 APIV 的数据分布类型是近似相互独立，HY-COCA 算法扫描了整个数据分布空间后，没有发现相关性差异足够显著的特殊相关性子区域。对于 APII 的数据分布，HY-COCA 在进行合并操作之后，返回了具有函数依赖的比较集中的 16 个子区域(子区域规模大于 20)。

结束语　本文将原来的检测数据间相关性的工作又推进了一步：发现数据分布中具有特殊相关性的子区域。在数据分布矩阵的基础上，利用相关系数这一准确描述属性间相关性的工具，解决了子区域搜索算法、子区域间相关性差别的度量以及如何将结果显示给用户等技术问题，提出了 HY-COCA 算法并分析了该算法的时间效率和空间效率。最后用生成的数据验证 HY-COCA 的有效性和效率，对于 HY-COCA 的有效性，定义了检出率和覆盖率两个指标，实验结果验证了 HY-COCA 算法的有效性，另外还用 HY-COCA 算法发现了 TPC-H 的数据集中属性间的相关性特征和具有特殊相关性的子区域。

在进行 HY-COCA 的工作中，还发现了一些问题，是未来工作中要解决的：1)效率问题，HY-COCA 的时间效率是 $O(n^3)$ 级别的，因此对于比较大的数据分布矩阵，精确的相关系数计算方法无法得到令人满意的时间效率，未来工作会考察采样方法或寻找大数据分布矩阵的近似算法；2)结果显示问题，目前 HY-COCA 是命令行界面的，找出的具有特殊相关性的子区域及其他特性是用坐标及其他数字形式显示给用户。未来工作会寻找用图形方法辅助检测数据相关性，这样对于用户来说更加直观，而且还可以方便与用户的交互。

参 考 文 献

[1] 王珊，曹巍，覃雄派. 基于熵相关系数的关联性自动判别方法——COCA[J]. 计算机应用，2006，26(9)：2005-2008
Wang Shan，Cao Wei，Qin Xiong-pai. COCA—a new way to auto-detect association based on entropy correlated coefficients [J]. Journal of Computer Applications，2006，26(9)：2005-2008

[2] 曹巍，王珊. 面向多维混合型数据分布的混合多维直方图初探[J]. 计算机应用，2009，29(9)：2487-2490
Cao Wei，Wang Shan. Exploration of hybrid multi-dimensional histograms for hybrid multi-dimensional data distribution[J]. Journal of Computer Applications，2009，29(9)：2487-2490

[3] 曹巍，王珊，覃雄派，等. 面向不同数据分布的多维直方图算法 COCA-Hist[J]. 计算机学报，2008，31(6)：1013-1024
Cao Wei，Wang Shan，Qin Xiong-pai，et al. Versatile Multidimensional Histograms for Different Data Distributions[J]. Chinese Journal of Computers，2008，31(6)：1013-1024

[4] Ilyas I F，Markl V，Haas P J，et al. CORDS：Automatic Discovery of Correlations and Soft Functional Dependencies[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Paris，France：ACM，2004：647-658

[5] Poosala V，Ioannidis Y. Selectivity Estimation Without the Attribute Value Independence Assumption[C]// Proceedings of 23rd International Conference on Very Large Data Bases. Athens，Greece：Morgan Kaufmann，1997：486-495

[6] Deshpande A，Garofalakis M. Independence is Good：Dependency-Based Histogram Synopses for High-Dimensional Data[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Santa Barbara，CA，USA：ACM，2001：199-210

[7] Lim L，Wang M，Vitter J S. SASH：A self-adaptive histogram set for dynamically changing workloads[C]//Proceedings of the 29th International Conference on Very Large Data Bases. Berlin，Germany：Morgan Kaufmann，2003：369-380

[8] Bruno N，Chaudhuri S，Gravano L. STHoles：A Multidimensional Workload-Aware Histogram[R]. Technical Report MSR-TR-2001-36

[9] 张尧庭，等. 定性资料的统计分析[M]. 桂林：广西师范大学出版社，1991：1-205
Zhang Yao-ting，et al. Statistical analysis of qualitative data [M]. Guilin：Guangxi Normal University Press，1991：1-205

[10] Poosala V，Haas P J，Ioannidis Y，et al. Improved Histograms for Selectivity Estimation of Range Predicates[C]// Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. Montreal，Quebec，Canada：ACM，1996：294-305

[11] Mueen A，Nath S，Liu J. Fast Approximate Correlation for Massive Time-series Data[C]// Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. Indianapolis，Indiana，USA：ACM，2010：171-182

[12] Moerkotte G，Neumann T，Steidl G. Preventing Bad Plans by Bounding the Impact of Cardinality Estimation Errors[J]. Proceedings of the VLDB Endowment，2009，2(1)：982-993

[13] Kanne C，Moerkotte G. Histogram Reloaded：the Merits of Bucket Diversity[C]// Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data. Indianapolis，

Indiana, USA: ACM, 2010: 663-674

[14] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to Algorithms[M]. Cambridge MA, USA: the MIT Press, 2009: 65

[15] Gunopulos D, Kollios G, Tsotras V, et al. Approximating multi-dimensional aggregate range queries over real attributes[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Dallas, Texas, USA: ACM, 2000: 463-474

[16] Aboulnaga A, Chaudhuri S. Self-tuning histograms: building histograms without looking at data[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Philadelphia, Pennsylvania, USA: ACM, 1999: 181-192

[17] Robinson J T. The K-D-B-Tree: A search structure for large multidimensional dynamic indexes[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Ann Arbor, Michigan, USA: ACM, 1981: 10-18

[18] Nievergelt J, Hinterberger H, Sevcik K C. The grid file: An adaptable, symmetric multikey file structure[J]. ACM Transactions on Database Systems, 1984, 9(1): 38-71

[19] Finkel R A, Bentley J L. Quad trees a data structure for retrieval on composite keys[J]. Acta Informatica, 1974, 4(1): 1-9

(上接第 192 页)

QEMU-SW 将每个 x86 访存操作对应的翻译代码内联到翻译块中，且在调用辅助 C 函数时有很多的 Spill 操作，因此 TB 的尺寸较大，平均每个 TB 达到 1144 字节(代码膨胀率为 67.4)。AB 和 AB-OPT 以辅助 C 函数的方式实现访存操作，并且特别规划了 Callee-saved 寄存器的使用，因此翻译块的平均大小分别为 279 字节和 255 字节，代码膨胀率降低到了 17.3 和 15.8。相比 AB 而言，虽然 AB-OPT 的代码膨胀率降低不多，但由于其中存在一些不会执行的补偿代码(主要是 store 操作)，因此实际代码膨胀率还要低一些。

本文在测试中仅考虑由于读取或保存 x86 寄存器所产生的 load/store 操作，访问虚拟机内存所引起的 load/store 操作不会被优化，因此不予考虑。QEMU-SW 孤立地将每条 x86 指令翻译为中间表示，即使后续指令使用了相同的寄存器，也要重新生成中间表示来将其由内存装入虚拟寄存器。另外，为支持精确异常，还需在每条源指令的翻译块的末尾生成 store 指令来将结果保存到源寄存器中。因此，QEMU-SW 翻译代码中的 load/store 数量较大，经过其自身的活跃性分析优化后，平均每个 TB 中的 load/store 数量分别为 5.8 和4.8。AB 的做法与 QEMU 类似，平均每个翻译块中的 load/store 数量为 5 和 4.4。由于 AB 对段级存储管理进行了优化，在计算访存地址时，若段基址为 0，则忽略其影响(测试表明，97.3%的访存指令符合此条件)，免去了从 cpu_state 区域装入段基址的操作，因此 AB 的 load 数量少于 QEMU。AB-OPT 中，平均每个翻译块中的 load/store 数量为 2.9 和 4.4，其中 store 的数量与 AB 基本持平，但其中一部分为补偿代码，并不真正执行，以 TLB 命中率为 99.4%进行计算，每个 TB 中的实际执行的 store 数量为 3.8，相比 QEMU、load/store 的数量分别降低了约 50%和 21%。

由于 QEMU-SW 采用三阶段翻译机制，因此翻译开销较大，每个 x86 指令字节的翻译时间(cycles/in-byte)为 2153 TICKS；AB 的翻译时间最短，其 cycles/in-byte 为 424 TICKS，但翻译代码质量不高；由于译码标注及优化翻译会引入一定的开销，因此 AB-OPT 的翻译时间有所增加，其 cycles/in-byte 为 920 TICKS，在翻译代码质量优于 QEMU-SW 的情况下，翻译开销降低了约 57%。

结束语 本文提出了一种译码制导的轻量级动态二进制翻译优化技术，在译码阶段提取源指令的高层语义信息，结合上下文对其进行标注，并在翻译阶段利用标注信息直接生成优化的目标指令。该技术可在不影响系统移植性的情况下，利用译码和翻译两个阶段即可识别动态二进制翻译系统中主要的基本块级优化机会，生成优化的本地代码，在降低动态二进制翻译开销的同时，提升翻译代码的质量。测试表明，相比 QEMU，采用该优化技术的 ARCH-BRIDGE 跨平台 x86 系统虚拟机的翻译开销降低了 53%，翻译块尺寸降低了 78%，load 和 store 操作数量分别了降低了 50%和 21%。

参 考 文 献

[1] Ebcioğlu K, Altman E R. DAISY: Dynamic compilation for 100% architectural compatibility[J]. ACM SIGARCH Computer Architecture News, ACM, 1997, 25(2): 26-37

[2] Hertzberg B, Olukotun K. DBT86: A Dynamic Binary Translation Research Framework for the CMP Era[C]// PESPMA 2009. 2009: 41-46

[3] Bala V, Duesterwald E, Banerjia S. Dynamo: a transparent dynamic optimization system[C]// ACM SIGPLAN Notices. ACM, 2000, 35(5): 1-12

[4] Bruening D, Qin Zhao, Amarasinghe S. Transparent Dynamic Instrumentation[J]. Sigplan Notices-SIGPLAN, 2012: 133-144

[5] Guan H B, Ma R H, Yang H B. MTCrossBit: A dynamic binary translation system based on multithreaded optimization[J]. Science ChinaInformation Sciences, 2011, 54(10): 2064-2078

[6] 包云程，梁阿磊，管海兵. 动态二进制翻译基础平台 CrossBit 的设计与实现[J]. 计算机工程，2007，33(23)：100-101
Bao Yun-cheng, Liang A-lei, Guan Hai-bing. Design and Implementation of CrossBit: Dyanmic Binary Translation Infrastructure[J]. Computer Engineering, 2007, 33(23): 100-101

[7] Bellard F. QEMU, a fast and portable dynamic translator[C]// USENIX annual technical conference, FREENIX Track. 2005: 41-46

[8] Payer M, Gross T R. Generating low-overhead dynamic binary translators[C]// Proceedings of the 3rd Annual Haifa Experimental Systems Conference. ACM, 2010: 22-36

[9] Sridhar S, Shapiro J S, Bungale P P. HDTrans: a low-overhead dynamic translator[J]. ACM SIGARCH Computer Architecture News, 2007, 35(1): 135-140

[10] Hu W, Wang J, Gao X. Godson-3: A scalable multicore RISC processor with X86 emulation[J]. Micro, IEEE, 2009, 29(2): 17-29

[11] 王荣华. 动态二进制翻译优化研究[D]. 杭州：浙江大学，2013
Wang Rong-hua. Research on Dyanmic Binary Translation Optimization[D]. Hangzhou: Zhejiang University, 2013

[12] 黄聪会，陈靖，龚水清，等. 64 位 Windows ABI 虚拟化方法研究[J]. 计算机科学，2014，41(1)：39-42
Huang Gong-hui, Chen Jing, Gong Shui-qing, et al. Research of Method for Virtualizing 64-bit Windows Application Binary Interface[J]. Computer Science, 2014, 41(1): 39-42