基于"C藤"Pair Copula的高维OLAP查询建模方法研究

倪志伟 王 超 高雅卓
(合肥工业大学管理学院智能管理研究所 合肥 230009)
(合肥工业大学过程优化与智能决策教育部重点实验室 合肥 230009)

摘 要 信息爆炸造成的数据仓库维度的急剧增加,大大影响了OLAP查询模型的精度和效率。首次将数理统计学中的"C藤"Pair Copula引入到OLAP查询建模的研究中,有效地解决了高维OLAP查询建模时的"维数灾难"问题,并设计了针对该模型的参数估计方法以提取数据概要知识。实验分析表明与传统方法相比,基于Pair Copula方法的模型可以在保证OLAP的查询精度的基础上减少数据立方体的存储空间,并且在高维数据环境下具有更高的查询效率。

关键词 OLAP近似查询,数据立方体,数据概要,Pair Copula,C藤
中图法分类号 TP311 **文献标识码** A

Efficient Modeling Method for Multidimensional OLAP Query Based on "C-Vine" Pair Copula

NI Zhi-wei WANG Chao GAO Ya-zhuo
(School of Management, Hefei University of Technology, Hefei 230009, China)
(Key Laboratory of Process Optimization and Intelligent Decision-Making, Ministry of Education, Hefei University of Technology, Heifei 230009, China)

Abstract The rapid increasing dimensionality of database caused by the recent information explosion greatly impairs the accuracy and efficiency of On-Line Analytical Processing(OLAP) query models. In this paper, by first applying the statistical concept "C-Vine" Pair Copula in the study of OLAP query model, an effective solution to the "dimension curse" of higher dimensional OLAP query models was provided, and a specific parametric estimation method was proposed to extract the data synopsis from the original mass data for those higher dimensional OLAP models. Experimental results show that compared with existing methods, the proposed Pair Copula-based model can reduce storage space for data cubes while improving the relatively high query accuracy of OLAP models, and especially it provides a better query efficiency for higher dimensional data cubes compared with existing methods.

Keywords OLAP approximate query, Data cube, Data synopsis, Pair copula, C-vine

1 引言

由于数据仓库中信息量十分巨大,因此在此基础之上如何实现对用户查询的快速响应以辅助决策过程,一直是研究者们十分关注的问题。从目前广为流行的商务智能系统来看,OLAP查询技术由于能够提供对原始信息的多维度描述,因此被广泛地用于系统查询模块的构建过程中,成为商务智能三大关键技术之一[1]。

通过OLAP查询,用户可以有效地从巨大的数据库中提取经过总结的信息,并参考这些信息从战略的角度去制定决策,从而摆脱底层数据的纷繁细节。然而随着技术的不断进步,信息爆炸使得用户面对的数据量越来越庞大,OLAP查询所耗费的时间成本也相应急剧增加,传统意义上的OLAP查询逐渐无法满足用户快速响应的要求。与此同时,研究者发现用户在进行OLAP查询时往往并不对具体维成员值所对应的精确数据感兴趣,而是更关注于数据所表现出来的趋势[2]。因此,越来越多的研究者开始通过提取数据立方体中的概要信息,为OLAP查询提供近似数据的方法来换取查询执行的效率。如文献[3]首先采用基于密度分层聚类方法将数据立方体划分成块,然后选用log-linear模型对每个数据块进行模型参数的计算,并存储模型参数用于OLAP近似查询;Y Chen等建立了一种回归数据立方体[4],在其基础上实现了立方体压缩,并对OLAP聚集操作提供近似值。除此之外,直方图技术、抽样技术、小波变换等技术也被运用到OLAP近似查询建模中。例如,V. Poosala等所提出的MHist直方图[5]以及D. Gunopulos等人提出的GENHist直方图[6]等都是经典的直方图方法;文献[7,8]等分别采用不同的抽样方法进行建模,以支持OLAP近似查询;而K. Chakrabarti等人则将小波变换这一数学工具运用到近似查询建模中来,将小波分解技术用于数据立方体,以获得由小波

到稿日期:2012-11-22 返修日期:2013-03-18 本文受国家863高技术研究发展计划基金项目(2011AA040501),国家自然科学基金项目(71271071,70871033)资助。

倪志伟(1963—),男,教授,博士生导师,主要研究方向为人工智能、数据挖掘、商务智能,E-mail:nzwgd@hfut.edu.cn;王 超(1983—),男,博士生,主要研究方向为动态数据挖掘、OLAP及商务智能;高雅卓(1984—),女,博士,主要研究方向为数据挖掘、OLAP及商务智能。

参数所组成的紧凑的数据概要[9]。

然而随着信息技术的快速发展和知识爆炸更加深化，数据库的庞大不仅仅体现在数据条目的急剧增加上，同时也体现在数据维度数的大幅增长。目前基于高维数据立方体的快速查询模型构建正遇到一个瓶颈，即现有的技术在处理高维度数据库时往往会陷入"维度灾难"的陷阱。以传统的相关性模型为例，在处理5维数据时需要估计 $5*(4)/2=10$ 个参数，处理20维数据时则将要估计100个参数。况且在高维度的情况下，大量参数的同时估计还会造成极大的估计误差[10]。因此目前大多数关于高维OLAP查询的研究都假设各维度之间是独立的(例如经典的GENHist算法等[6])，这就不可避免地会产生较大的查询误差。

Copula函数也称为连接函数，其概念最早由Sklar于1959年提出[11]，它是利用样本数据的边缘分布来近似确定其联合分布的函数，是在构造多元联合分布以及随机变量间相关结构分析中的常用工具。目前Copula方法已在金融、行为分析等多个领域得到广泛的应用[12,13]。在数据挖掘领域，我们曾将Copula方法引入到OLAP查询建模过程中[14]，取得了较好的效果。然而传统的Copula模型由于假设两两维度之间的相关结构相同，因此当OLAP数据集维度较高、维度间差异较大时可能会带来较大的建模误差，从而影响查询的精确性。Aas等[15]首次系统地研究了基于"藤"结构的Pair Copula构造方法，为高维模型构造打开了一个新局面。在总结以往研究的基础上，本文首次将基于"藤"结构的Pair Copula方法从统计领域引入到OLAP快速查询建模的领域中，以提高高维OLAP查询模型的速度和准确率。

本文所采用的基于"藤"结构的Pair Copula有如下优点：(1)适用于高维数据集，易于参数估计，且估计误差较小。由于在"C藤"Copula中维度增加时仅仅是在"藤"上新加入一个分枝，并不会大幅增加运算的时间，因此该模型几乎不受维数灾难影响。(2)模型构造灵活，当数据集中增加新的维度时只需要估计新的"分叉"即可，不需要重新估计模型。(3)可以全面捕捉维度间的线性与非线性关系，对较常用的相关系数来说更为准确。(4)与传统Copula函数相比，基于"藤"结构的Pair Copula可以根据样本数据的特征自由选取和构造其相关结构，模型对于数据联合分布的拟合结果也更加精确，从而可以减小查询误差。此外，本文所建立的统计模型提供了直接在连续属性上进行OLAP查询的能力，而不用像传统的OLAP查询方法那样事先对连续属性进行离散化[16]。

2 相关概念

2.1 OLAP数据立方体及OLAP聚集函数

OLAP查询以数据立方体为数据基础。数据立方体是由多个维度组成的数据集合，其中的维度包括观察维度和度量维度。每个观察维度又可以按照不同的粒度划分为不同的层次。OLAP查询所涉及的上卷、下钻、切片、切块等操作就是在数据立方体观察维度这种多维和分层的结构上进行的；而对度量维度值的计算，则通常根据不同的需要，通过一个或几个数值函数进行计算。

定义1(数据立方体Data Cube) 将数据立方体表示为一个五元组 $DC=\langle D,H,g,M,f\rangle$。5个元素含义如下：

(1)$D=\{X_1,X_2,\cdots,X_n\}$为观察维度的集合，其中 n 表示观察维度的个数；

(2)$H=\{H_1,H_2,\cdots,H_k\}$表示观察维度可以划分的层次，其中 k 表示层次数；

(3)$g:D\rightarrow H$ 表示一个一对多的映射。$g(X_i)=\{H_1,\cdots,H_t\}$表示某观察维度 X_i 包含了 t 个层次；

(4)$M=\{M_1,M_2,\cdots,M_m\}$为度量维度的集合，其中 m 为度量维度的个数。它与观察维度集合 D 满足：$D\cap M=\emptyset$；

(5)f 为作用在度量维度 M 上的聚集函数。

数据立方体中度量维度集合 M 中各个对象的取值一般都是数值形式，可以通过一些数值函数进行各种运算，即定义1中的 f 可以表示多种聚集函数。如，当需要得到 $M_j(M_j\in M)$沿某个观察维度 $X_i(X_i\in DIM)$的聚集值，通常使用 *sum*()函数对度量维度进行求和运算。而常用的度量维度上的聚集数值函数通常可以分为分布的和代数的。

所谓分布式(distributive)函数，指的是该聚集函数能通过分布式方式计算得到。设数据划分为若干个集合，将某函数用于每个划分，得到若干个聚集值，若所得的聚集值之和与将该函数用于整个数据集得到的结果一样，就说该函数可以用分布式方法计算。例如，数据立方体聚集计算中频繁出现的 *count*()，*sum*()都属于分布式聚集函数。

代数型(algebraic)函数，即指该函数能够用具有有限个参数的代数函数计算，而每个参数都可以用一个分布式聚集函数求得。如数据立方体聚集计算中经常用到的 *average*()或 *mean*()函数可以通过 *sum*()/*count*()计算，其中 *count*()，*sum*()都是分布式聚集函数。

以上所述 *count*()，*sum*()和 *average*()都是OLAP近似查询的重要研究对象[17]，也正是本文的研究重点。

2.2 Copula方法简介

Copula是一种从多维随机向量联合分布中提取相依结构的数学方法。

定义2(Copula函数) d 维Copula函数 C 是指具有以下性质的一类多元函数：

(1)$C:[0,1]^d\rightarrow[0,1]$；

(2)C 对它的每个变量都是单调递增的；

(3)$C(u_1,\cdots,u_{k-1},0,u_{k+1},\cdots,u_d)=0$ 且 $C(1,\cdots,1,u_k,1,\cdots,1)=u_k$，其中 $u_1,\cdots,u_d\in[0,1]$。

显然，从以上定义可以看到若有 p 维随机向量$(X_1,X_2,\cdots,X_p)$，其联合分布是 $F(X_1,X_2,\cdots,X_p)$，边缘分布分别是 $F_1,F_2,\cdots,F_p$，C 是随机向量的Copula函数，那么根据定义1，p 维随机向量的联合分布可以通过Copula函数写成下式：

$$F(x_1,x_2,\cdots,x_p)=C(F_1(x_1),F_2(x_2),\cdots,F_p(x_p)) \quad (1)$$

由于 $F_i(x_i)$也是一个随机变量并服从$[0,1]$均匀分布，因此函数 C 也是一个分布函数。单就这个函数本身来看，它不含有任何边缘分布信息，而是提取了各个随机变量之间的相依结构。其在OLAP查询中也就是抓住了各个维度之间的关联关系，从而有助于建立一个整体的查询模型。

目前关于Copula函数 C 的选择有很多种，包括Gauss、t以及Archimedean Copula族等等。但它们往往仅能较好地应用于二元或者同结构多元的情况，对于OLAP这种结构复杂的数据集会造成较大误差。为此下面引入"C藤"Pair Copula的概念。

设 $f(x_1,x_2,\cdots,x_p)$为联合分布 $F(X_1,X_2,\cdots,X_p)$的密度函数，则由贝叶斯公式可以得到密度函数的分解形式：

$$f(x_1,x_2,\cdots,x_p)=f(x_1)\cdot f(x_2|x_1)\cdot f(x_3|x_1,x_2)\cdots$$

$$\cdot f(x_p|x_1,\cdots,x_{p-1}) \quad (2)$$

根据 Copula 函数的定义，式(2)可以进一步表示为 Copula 函数的形式。以 3 维为例，设 c_{12} 为 x_1 和 x_2 的 Copula 函数，则有：

$$f(x_2|x_1)=c_{12}(F_1(x_1),F_2(x_2))f_2(x_2) \quad (3)$$

同理有：

$$f(x_3|x_1,x_2)=c_{23|1}(F_{2|1}(x_2|x_1),F_{3|1}(x_3|x_1))c_{13}(F_1(x_1),F_3(x_3))f_3(x_3) \quad (4)$$

所以有：

$$f(x_1,x_2,x_3)=f_1(x_1)f_2(x_2)f_3(x_3)\cdot c_{12}(F_1(x_1),F_2(x_2))\cdot c_{13}(F_1(x_1),F_3(x_3))\cdot c_{23|1}(F_{2|1}(x_2|x_1),F_{3|1}(x_3|x_1)) \quad (5)$$

以上即为 3 维情况下基于“C 藤”Copula 的密度函数分解，推广到多元的形式如下：

$$c(F_1(x_1),\cdots,F_p(x_p))=\prod_{j=1}^{p-1}\prod_{i=1}^{p-j}c_{j,j+i|1,\cdots,j-1}(F(x_j|x_1,\cdots,x_{j-1}),F(x_{j+i}|x_1,\cdots,x_{j-1})) \quad (6)$$

引理 1[18] 设 $u_1=F(x_1|y)$，$u_2=F(x_2|y)$ 为条件分布函数，且 $F(x_1,x_2|y)=C(u_1,u_2;\theta)$。其中 C 是一个参数为 θ 的二元 Copula 函数，那么有 $F(x_1|x_2,y)=h(u_1|u_2;\theta)$。其中：

$$h(u_1|u_2;\theta)=\frac{\partial C(u_1,u_2;\theta)}{\partial u_2}$$

根据引理 1，式(6)中的条件分布函数可以通过下面公式计算得到：

$$F(y|v)=\frac{\partial C_{y,v_j|v_{-j}}(F(y|v_{-j}),F(v_j|v_{-j}))}{\partial F(v_j|v_{-j})} \quad (7)$$

式中，v_{-j} 表示向量 v 中除去元素 v_j 后剩余的部分。从图 1 中可以更加清楚地看出基于“C 藤”结构的密度函数分解过程。不难发现“C 藤”Copula 密度函数分解的特点在于以一个关键变量为核心，层层深入。这一特点恰恰与 OLAP 数据立方体的结构是相吻合的，在实际运用中可以把 OLAP 的度量维度看作是关键结点。

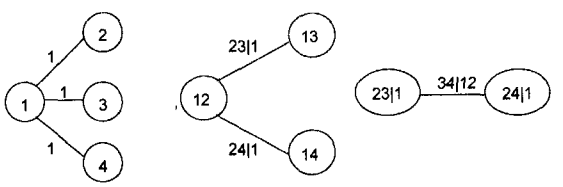

图 1 基于“C 藤”结构 Copula 的密度函数分解图

3 基于“C 藤”Copula 的 OLAP 查询模型构建

3.1 “C 藤”Copula 模型选择

在实际应用中考虑到如果将 OLAP 数据立方体按照“C 藤”结构完全分解会花费大量的时间和空间，因此根据 Heinen等的思想[10]，这里采用一种相对简化的分解方法。以 5 维为例，密度函数可以分解为：

$$f(m,x_1,x_2,x_3,x_4)=f(m)\cdot f(x_1)\cdot f(x_2)\cdot f(x_3)\cdot f(x_4)\cdot c_{m,x_1}(F(m),F(x_1))\cdot c_{m,x_2}(F(m),F(x_2))\cdot c_{m,x_3}(F(m),F(x_3))\cdot c_{m,x_4}(F(m),F(x_4))\cdot c_{x_1,x_2,x_3,x_4|m}(F(x_1|m),F(x_2|m),F(x_3|m),F(x_4|m)) \quad (8)$$

由式(8)可以看出，联合分布的密度函数可以分解为各维度的边缘分布、各观察维度和度量维度的二元 Copula 函数，以及已知度量维度值的条件下各观察维度之间的多元 Copula 函数 3 者的乘积。

要建立类似式(8)所示的多元 Pair Copula 模型，首先需要拟合各维度的边缘分布。由于在 OLAP 数据立方体中常常出现维度数据无法用一种已知分布来拟合的情况，本文根据高雅卓等的研究成果[14]，使用核密度估计来拟合各维度的边缘分布。设 X_i 为一个观察维度，则根据核密度估计的定义，其概率密度函数的核估计可以表示为：

$$\hat{f}(x_i)=\frac{1}{lh^d\det(S)^{1/2}}\sum_{j=1}^{n}K\left[\frac{(x_i-x_{ij})^TS^{-1}(x_i-x_{ij})}{h^2}\right] \quad (9)$$

式中，$K(\cdot)$ 为核函数，本文采用 Gauss 核函数。h 为窗宽系数，l 为样本容量。S 为 X_i 的 $d\times d$ 维协方差矩阵。

考虑到不同观察维度和度量维度的相关性也不相同，在实际操作中可以选用 Gaussian Copula、Archimedean Copula 等多种模型来拟合其相关结构，从而提高查询精度。

式(8)中的多元 Copula，本文采用 Gaussian Copula 来替代，其具体函数形式如式(10)所示。

$$C_\Gamma^{Normal}(u_1,\cdots,u_n)=\Phi_{\Gamma,n}(\phi^{-1}(u_1),\cdots,\phi^{-1}(u_n)) \quad (10)$$

式中，Γ 为多维正态分布中的相关系数矩阵，$\Phi_{\Gamma,n}$ 为相关系数矩阵为 Γ 的 n 维正态分布函数，ϕ^{-1} 为标准正态分布函数的反函数。u_i 表示 x_i 的边缘分布 $F_i(\cdot)$，$i=1,\cdots,n$。

综合式(8)－式(10)，即可得到 OLAP 数据立方体各维度的联合分布函数，进而可以满足 OLAP 的各种查询和聚集计算要求。

3.2 模型参数估计

由于“C 藤”Copula 模型比较复杂，本文在 Patton 提出的“两阶段参数估计方法”[19]的基础上设计了“分布参数估计法”对模型的参数进行估计。设度量维度 $X_M=\{x_{M,t}\}_{t=1}^T$，观察维度 $X_i=\{x_{i,t}\}_{t=1}^T$，$i=1,\cdots,n$。$X=(X_M,X_1,X_2,\cdots,X_n)$，则“C 藤”模型的对数极大似然函数 $L(X,\Omega)$ 可以写成如下形式：

$$L(X,\Omega)=L_1(X)+L_2(X;\Theta_M)+L_3(X;\Theta_M,\Theta_G) \quad (11)$$

$$L_1(X)=\sum_{t=1}^{T}(\log(f(x_M))+\sum_{i=1}^{n}\log(f(x_i))) \quad (12)$$

$$L_2(X;\Theta_M)=\sum_{t=1}^{T}\sum_{i=1}^{n}\log(c_{M,x_i}(F(x_{M,t}),F(x_{i,t}));\Theta_M) \quad (13)$$

$$L_3(X;\Theta_M,\Theta_G)=\sum_{t=1}^{T}\log(c_{G|M,x_1,\cdots,x_T}(\cdot;\Theta_G)) \quad (14)$$

式中，$\Theta_M=(\theta_{M,1},\theta_{M,1},\cdots,\theta_{M,n})$ 为二元 Copula 函数的参数，Θ_G 为多元 Gaussian Copula 函数的参数。由于边缘分布采用的是核密度估计，不需要参数估计，因此最终的参数集合为 $\Omega=(\Theta_M,\Theta_G)$。根据“分布参数估计法”的思想，首先估计出 $\hat{\Theta}_M$ $=\text{argmax}\ L_2(X;\Theta_M)$，接着通过 $\hat{\Theta}_M$ 得到 $\hat{\Theta}_G=\text{argmax}\ L_3(X;\hat{\Theta}_M,\Theta_G)$。

3.3 OLAP 查询

执行 OLAP 查询操作，实际上是对各种聚集函数进行求解的过程。本文建立 OLAP 查询模型，主要针对常用的聚集函数类型，即分布的和代数的两种类型。若要求查询满足一定维度值范围的记录条数，假设查询条件中包含 k 个维度，分别记为 $\{X_1,X_2,\cdots,X_i,\cdots,X_k\}$，其中 $k\leqslant n$，维度 X_i 的要求查询范围为 $[a_i,b_i]$，则未包含在查询条件中的维度可记为 $\{X_{k+1},X_{k+2},\cdots,X_n\}$。在这种情况下，满足查询条件的记录

条数就可以通过下式进行计算：

$$COUNT = T \cdot \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{a_1}^{b_1} \cdots \int_{a_k}^{b_k} f^*(x_M, x_1, x_2, \cdots, x_n) \mathrm{d}x_k \cdots \mathrm{d}x_1 \mathrm{d}x_{k+1} \cdots \mathrm{d}x_n \mathrm{d}x_M \quad (15)$$

式中，$f^*(x_M, x_1, x_2, \cdots, x_n)$为 n 个随机变量的联合分布函数。由于维数较高时会对式(15)的积分花费大量时间，为了提高算法的运行效率，由式(8)可以将式(15)改写为如下形式：

$$COUNT = T \cdot \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{a_1}^{b_1} \cdots \int_{a_k}^{b_k} f(m) \cdot f(x_1) \cdots f(x_n) \cdot c_{m,x_1}(F(m), F(x_1)) \cdots \cdot c_{m,x_n}(F(m), F(x_n)) \cdot c_{x_1,\cdots,x_n|m}(F(x_1 \mid m), \cdots F(x_n \mid m)) \mathrm{d}x_k \cdots \mathrm{d}x_1 \mathrm{d}x_{k+1} \cdots \mathrm{d}x_n \mathrm{d}x_M \quad (16)$$

可以看到，式(16)中的被积函数是由一系列一元、二元函数和一个多元正态函数相乘构成的。根据 3.2 节的方法估计出模型参数后，式(16)的积分可以通过 Monte Carlo 方法计算出来，从而得到满足查询条件的记录个数。若需要查询满足一定维度取值范围条件的记录对应度量维度值的和，即聚集值，则与以上计数查询类似，设查询条件中包含 k 个维度，度量维度为 X_M。可以使用下式进行求和计算：

$$SUM = T \cdot \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \int_{a_1}^{b_1} \cdots \int_{a_k}^{b_k} x_M \cdot f^*(x_M, x_1, x_2, \cdots, x_n) \mathrm{d}x_k \cdots \mathrm{d}x_1 \mathrm{d}x_{k+1} \cdots \mathrm{d}x_n \mathrm{d}x_M \quad (17)$$

同式(16)一样，式(17)的积分也可以通过替代被积函数 f^* 的方式化简。当 OLAP 查询涉及到平均值查询时，即需要求得聚集值与计数值之间的比值，这时可通过公式 $AVG = SUM/COUNT$，在所涉及的聚集值与计数值的基础之上直接得到查询结果。从模型建立过程中可以看出，该模型能够通过变换积分上下限，灵活且有效地支持数据立方体观察维度 D 所对应的各层次 H 之间的上卷、下钻操作；同时切块、切片等操作也可以通过对积分目标的选取进行有效实现。

在模型构建过程中所涉及到的主要算法描述如下：

输入：t×n 维样本数据矩阵 Data，其中第一列为度量维度；查询矩阵 Query，存储用户对各维度的查询范围；已估计出的模型参数集合 $\Omega = (\Theta_M, \Theta_G)$，模拟次数 M。

输出：COUNT 变量，记录查询范围内的记录数。

过程：

Step 1：

Set COUNT=0；

[t,n]=size(Data)；%获取数据矩阵的行数和列数

Step 2：

W=random(n)；%生成 n 个服从[0,1]均匀分布的随机数

Step 3：

FOR Data 中的每一个维度 i∈[1,2,…,n]

IF i=1

$F_1 = W(1)$；

ELSE IF i<n

$F_i = C_{i|1}^{-1}(W(i) \mid F_1, \Theta_M)$；

ELSE

$F_i = C_{n|1,\cdots,n-1}^{-1}(W(i) \mid F_{2|1}, \cdots, F_{n-1|1}, \Theta_G)$；

END %生成服从 Pair Copula 分布的随机数，其中 C 为 Copula 函数，其条件分布可通过式(7)计算得到。

Step 4：

X_1=invcdf(F)；%通过核密度估计分别计算分布函数反函数的值，也即各个维度的 Monte Carlo 模拟值。

Step 5：

重复 M 次步骤 Step2－Step4，将结果存储在 M×n 维矩阵 X 中。

Step 6：

RETURN COUNT=t * SUM(X,IF($X_i \in$Query))/M。

4 实验分析

4.1 实验数据与实验平台

文章分别采用不同的分布和参数模拟生成了 4 组实验数据来检验本文算法的效果。实验数据集基本包括了不同维度关联程度、不同维度数以及不同记录数等各种情形。各数据集的具体情况如表 1 所列。为了便于比较，本文同时实现了传统的 Copula 算法以及目前较为流行的基准算法 GENHist 在以上几个实验数据集上的运行结果。

表 1 实验数据集设定

数据集	分布种类	相关系数设定区间	包含维度数	包含记录数
1	Gauss 分布	[0.3,1]	7	1000000
2	Gauss 分布	[0,0.3]	7	1000000
3	Gauss 分布	[0.3,1]	15	500000
4	t 分布	[0.3,1]	7	1000000

本文的实验环境为 Intel Core 2 Duo 2.00GHz，2G RAM，操作系统为 Windows XP。实验过程中使用 SQL Server 2005 作为数据存储工具，算法使用 Matlab 2010b 编程实现。

4.2 实验结果及分析

根据第 3 节关于“C 藤”Pair Copula 模型的构建方法，首先采用核密度估计对各维度样本数据进行拟合，其中窗宽由交叉鉴定法确定。接着再分别对每一对 Copula 进行选择和参数估计，最后得到各维度的联合分布密度函数 f^*。由 3.3 节的算法即可得到相应查询值。为了更加全面地测试“C 藤”Pair Copula 模型的运行效果，本文针对每个数据集生成了 2 组查询来模拟查询，每组查询均为随机生成。2 组查询分别覆盖整个数据集的 10%和 5%。下面分别从查询精度、查询时间和存储空间等方面来分析“C 藤”Pair Copula 模型的表现。除非特殊说明，所有实验均随机抽取数据集记录总数的 1%进行核密度估计，Monte Carlo 模拟次数取 2000 次。

(1)查询精度比较

设 $COUNT^*(Data, Query)$ 和 $COUNT(Data, Query)$ 分别表示查询 $Query$ 在 OLAP 数据集 $Data$ 上计数的真实值和计算值，则查询的相对平均误差可以定义为：

$$\varepsilon(Data, Query) = \frac{1}{k} \sum_{i=1}^{k} \frac{|COUNT^*(Data, Query_i) - COUNT(Data, Query_i)|}{\max(1, COUNT^*(Data, Query_i))} \quad (18)$$

本文首先通过第 1 组查询来比较 Pair Copula 算法、Gaussian Copula 算法和经典 GENHist 算法的平均查询精度。文章随机生成了第 1 组查询 100 次，分别在表 1 中的第 1、第 2 以及第 4 个数据集上执行。其中 GENHist 算法的初始划分参数设定为 15。所得的平均查询误差如图 2 所示。

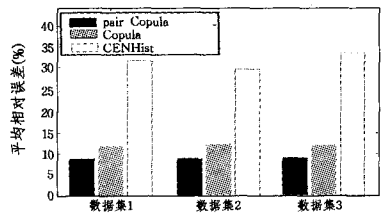

图 2　不同数据集下 3 种算法的平均查询误差比较

由图 2 可以看出 Pair Copula 算法在 3 个数据集上的平均查询误差都是最低的，而 GENHist 算法在 7 维数据集上的表现难以令人满意，其平均误差与低维度数据集相比有较大幅度的上升。此外，从图中可以看到 Pair Copula 算法在第 1 个和第 4 个数据集上的查询精度要高于第 2 个数据集，而 GENHist 算法的结果则与之相反。这是因为第 1 个和第 4 个数据集所包含维度之间的相关性较高，若忽略这种相关性则会对查询精度产生影响。为了更进一步比较数据维度变化对查询精度的影响，本文以数据集 3 为研究对象分别比较了 3 种算法在不同数据维度下的表现。实验中每次查询的维度均为随机抽取。由图 3 可以看出当数据集的维度逐渐升高到 10 维时 GENHist 算法基本已经失效了，这主要是由于 GENHist 算法在拟合各个“桶”之间的联合分布时采用的是多元核密度估计，在维度较高时查询的精度会大幅下降。相对来看传统的 Copula 算法和 Pair Copula 算法的查询平均误差并没有随着数据集维度的升高大幅增加，但 Pair Copula 算法的平均误差上升的速度要更加缓于 Copula 算法，说明 Pair Copula 算法具有更高的维度稳定性。

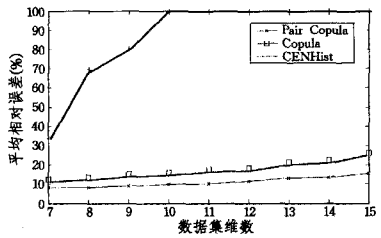

图 3　不同数据维度下 3 种算法的平均查询误差变化

下面以数据集 1 为例，讨论 Monte Carlo 模拟次数与 OLAP 查询精度的关系。如图 4 所示，Pair Copula 模型的查询平均误差随着模拟次数的增加而不断下降，但下降的幅度越来越慢。实际运用模型时可以根据用户需求灵活调整模拟次数来寻求查询时间(查询时间与模拟次数基本呈线性关系)和精度的平衡。

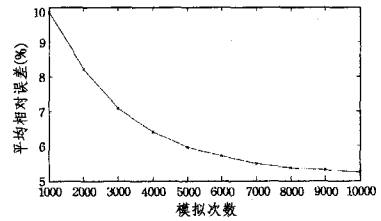

图 4　不同模拟次数下 Pair Copula 模型的平均查询误差

(2)查询时间比较

下面通过比较查询时间来分析不同算法的运行效率。这里 Gaussian Copula 和 Pair Copula 算法的查询时间指模型参数估计完成后对于用户每次查询的响应时间；GENHist 算法的查询时间同样是指计算出各个“桶”之后进行多元核密度估计的时间。图 5 给出了不同算法在数据集 1 上分别使用 2 组查询得到的查询时间对比。

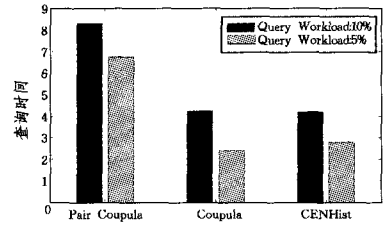

图 5　不同查询覆盖范围下 3 种算法的平均查询时间

由图 5 可以看到 Pair Copula 方法对于查询的响应时间要慢于其它两种算法，这主要是由于 Pair Copula 算法采用 Monte Carlo 方法来计算积分，在低维度下效率要低于其它两种算法，但时间仍然低于在相同的条件下使用 SQL 语句直接进行两组查询所花费的 14.63s 和 8.59s。值得注意的是，也正是由于 Pair Copula 方法采用了该积分计算方法，使得当查询覆盖范围增加时其额外时间耗费要远低于其它两种算法。即当用户查询涉及的记录条数比较多时，Pair Copula 算法的效率优势将逐渐显现。经测试，当查询覆盖范围由 10%上升到 20%时，Pair Copula 方法的查询时间仅上升了约 25%至 10.33s，而 Copula 和 GENHist 算法的查询时间则分别上升了近 60%和 73%，达到 6.81s 和 7.25s。

实验基于数据集 3 以及第 1 组查询考查了不同算法的查询时间随数据维度变化的情况，结果如图 6 所示。从图 6 中可以看到当数据维度上升到 10 维时，Pair Copula 算法的效率已经超过了传统的 Copula 算法，且其查询时间并不会随着维度的增加而大幅上升。而 GENHist 算法由于并不考虑高维相关性，因此查询时间相对较短。

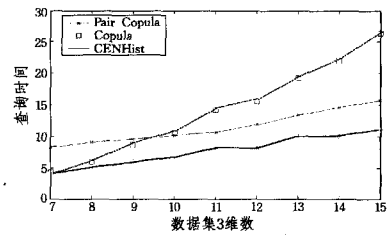

图 6　不同数据维度下 3 种算法的平均查询时间(基于数据集 3 和第 1 组查询)

(3)存储空间比较

核密度估计是一种非参数估计，其估计精度与样本量是密切相关的。故当为 Pair Copula 算法分配较多存储空间时也会相应提高算法的精确性。图 7 给出了不同存储空间下 Pair Copula 算法的平均误差变化，其中计算样本均为随机抽取。

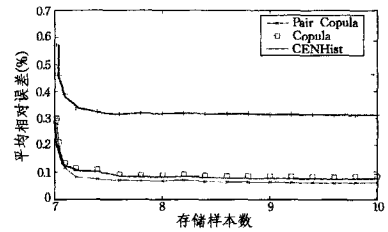

图 7　不同存储样本数目下 3 种算法的平均查询误差

从图 7 中可以看到随着存储样本个数的增加，3 种模型查询的平均误差先是快速下降然后逐渐趋于平缓。这说明在一定误差范围下并不需要使用所有样本来进行核密度估计，从而节省了存储空间。

此外，从图中还可以看出与其它算法相比，Pair Copula 算法能在较少的存储空间下达到较高的查询精度。

结束语 由于数理统计中的 Pair Copula 方法可以准确快速地拟合高维随机变量的联合分布，且具有易于参数估计、模型构造灵活、可以全面捕捉维度间的线性与非线性关系等特点，本文首次将其引入到 OLAP 查询中，建立了基于“C 藤” Pair Copula 的高维 OLAP 查询模型。首先针对 OLAP 查询的特点设计了相应的 Pair Copula 查询模型，并同时给出了较为快捷的模型参数估计方法；接着结合 OLAP 操作的特点具体分析阐述了使用 Pair Copula 模型进行 OLAP 查询的具体过程，并给出了相应的算法；最后为了证明该算法的有效性，基于 4 组数据设计了对比试验，将本文提出的 Pair Copula 算法与经典 GENHist 算法、传统 Copula 算法进行了比较。实验结果显示相对于其它两种算法，本文提出的 Pair Copula 算法在同等条件下具有较低的平均查询误差以及较高的空间使用率，并且当数据维度较高时其查询效率要明显高于其它两种算法。

本文的研究是在传统 Copula 查询算法[14]上的进一步改进，取得了令人满意的研究结果。但仍然存在许多问题有待更深入的探讨。首先在数据挖掘方面，由于在使用 Copula 方法建立 OLAP 查询模型的同时建立了各维度样本间的联合分布，因此我们可以借此对维度间的关系进行挖掘，实现 OLAP 技术与数据挖掘的结合。其次，由于 Copula 方法同样可以用来拟合离散变量的联合分布，在今后的工作中，我们可以以此为出发点进行进一步的研究与实验，进一步扩大基于 Pair Copula 的 OLAP 建模方法的使用范围。

参 考 文 献

[1] Chaudhuri S, dayal U, Narasayya V. An overview of business intelligence technology [J]. Communications of the ACM, 2011, 54(8): 88-98

[2] Cuzzocrea A. Improving range-sum query evaluation on data cubes via polynomial approximation [J]. Data & Knowledge Engineering, 2006, 56(2): 85-121

[3] Barbará D, Wu X T. Loglinear-Based Quasi Cubes [J]. Journal of Intelligent Information Systems, 2001, 16(3): 255-276

[4] Chen Y, Dong G, Han J W, et al. Regression Cubes with Lossless Compression and Aggregation [J]. IEEE Transations on Knowledge and Data Engineering, 2006, 18(12): 1585-1599

[5] Poosala V, Ioannidis Y E. Selectivity estimation without the attribute value independence assumption [C]// Proceedings of the 23rd International Conference on Very Large Databases. Athens, Greece, August 1997: 486-495

[6] Gunopulos D, Kollios G, Tsotras V J, et al. Approximating Multi-Dimensional Aggregate Range Queries Over Real Attributes [C]// Proceedings of the 2000 ACM SIGMOD international conference on Management of data. Dallas, Texas, USA, May 2000: 463-474

[7] Rösch P, Lehner W. A Sample Advisor for Approximate Query Processing [C]// Proceedings of the 14th east European conference on Advances in databases and information systems. Novi Sad, September 2010: 490-504

[8] Li Xiao-lei, Han Jia-wei, Yin Zhi-jun, et al. Sampling cube: a framework for statistical OLAP over sampling data [C]// Proceedings of the 2008 ACM SIGMOD international conference on management of data. Vancouver, BC, Canada, June 2008: 779-790

[9] Chakrabarti K, Garofalakis M, Rastogi R, et al. Approximate Query Processing Using Wavelets [J]. The International Journal on Very Large Data Bases, 2001, 10(2/3): 199-223

[10] Heinen A, Valdesogo A. Asymmetric CAPM dependence for large dimensions: the canonical vine autoregressive model[M]. CORE discussion papers 2009069, Universit _ ecatholique de Louvain, Center for Operations Research and Econometrics (CORE), 2009

[11] Sklar A. Fonctions de répartition à n dimensions et leurs marges [M]. Publications de l'Institut de Statistique de l'Universite de Paris 8, 1959: 131-229

[12] Aas K, Berg D, Kurowicka D. Modeling Dependence Between Financial Returns Using Pair-Copula Constructions [M]. Dependence Modeling: Vine Copula Handbook. World Scientific, 2011: 305-328

[13] Bhat C R, Eluru N. A copula-based approach to accommodate residential self-selection effects in travel behavior modeling [J]. Transportation Research Part B: Methodological, 2009, 43(7): 749-765

[14] 高雅卓，倪志伟，倪丽萍. 连续属性上的 OLAP 查询建模方法研究[J]. 情报学报，2011，30(4)：372-379

[15] Aas K, Czado C, Frigessi A, et al. Pair-copula constructions of multiple dependence [J]. Insurance: Mathematics and Economics, 2009, 44(2): 182-198

[16] Shanmugasundaram J, Fayyad U, Bradley P S. Compressed Data Cubes for OLAP Aggregate Query Approximation on Continuous Dimensions [C]// Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. San Diego, CA, USA, Aug. 1999: 223-232

[17] Acharya S, Gibbons P B, Poosala V, et al. The AQUA approximate query answering system [C]// Proceedings of the 1999 ACM SIGMOD international conference on Management of data. Philadelphia, Pennsylvania, USA, June 1999: 574-576

[18] Joe H. Families of m-variate distributions with given margins and m(m−1)/2 bivariate dependence parameters[J]. Lecture Notes-Monograph Series, 1996, 28: 120-141

[19] Patton A. Estimation of multivariate models for time series of possibly different lengths[J]. Journal of Applied Econometrics, 2006, 21(2): 147-173

(上接第 135 页)

[8] Itoh M, Chua L O. Advanced image processing cellular neural networks[J]. International Journal of Bifurcation and Chaos, 2007, 17: 1109-1150

[9] Itoh M, Chua L O. Memristor Cellular Automata and Memristor Discrete-Time Cellular Neural Networks[J]. International Journal of Bifurcation and Chaos, 2009, 19(11): 3605-3656

[10] Adamatzky A, Chua L O. Memristive excitable cellular automata [J]. International Journal of Bifurcation and Chaos, 2011, 21(11): 3083-3102

[11] 臧鸿雁，闵乐泉，吴春雪，等. 基于离散混沌系统广义同步定力的数字图像加密方案[J]. 北京科技大学学报，2007，29(1)：96-101

[12] 朱从旭，陈志刚，欧阳文卫. 一种基于广义 Chen's 混沌系统的图像加密新算法[J]. 中南大学学报：自然科学版，2006，37(6)：1142-1148

[13] Islam N, Puech W. Decryption of noisy encrypted images by statistical analysis[C]// 2011 3rd European Workshop on Visual Information Processing(EUVIP). Paris, 2011, 192-198