第 36 卷 第 5 期 计 算 机 工 程 2010 年 3 月
Vol.36 No.5 Computer Engineering March 2010
·软件技术与数据库· 文章编号：1000—3428(2010)05—0064—03 文献标识码：A 中图分类号：TP391.72

基于分布密度的直方图与选择率估计

朱 亮[1]，冯彦超[1]，刘椿年[2]，杨文柱[1]
(1. 河北大学数学与计算机学院河北省机器学习与计算智能重点实验室，保定 071002；2. 北京工业大学计算机学院，北京 100022)

摘 要：查询选择率估计是查询处理和优化中的关键之一。提出一种基于区域分布密度的方法，用于构造直方图，使其每个桶具有均匀分布或近似均匀分布，利用直方图估计查询选择率。实验结果表明，该方法对低维数据估计得到的查询选择率精度较高，并能对高维数据进行估计。

关键词：选择率估计；直方图；n 维超矩形；分布密度

Histogram Based on Distribution Density and Selectivity Estimation

ZHU Liang[1], FENG Yan-chao[1], LIU Chun-nian[2], YANG Wen-zhu[1]
(1. Hebei Province Key Laboratory of Machine Learning and Computational Intelligence, School of Mathematics and Computer Science, Hebei University, Baoding 071002; 2. College of Computer Science, Beijing University of Technology, Beijing 100022)

【Abstract】 Query selectivity estimation is one of the key issues for query processing and optimization. This paper presents a method based on domain distribution density to establish histograms in which the distribution of buckets is uniform or nearly. It utilizes the histograms to estimate query selectivity. Experimental results indicate that this method gets query selectivity with high precision for low-dimensional data and can estimate high-dimensional data.

【Key words】 selectivity estimation; histogram; n-dimension hyperrectangle; distribution density

1 概述

查询优化的关键步骤是查询选择率的估计。查询选择率估计的精度直接影响最优计划的选取，已有许多查询选择率估计方法，但多数方法在每次收集统计数据时，需要额外的 I/O 操作访问数据库，导致开销巨大，只能脱机执行或在负载较轻的系统中执行。很多方法只对低维数据有效，例如，当数据不超过 3 维时，某些直方图方法是有效的，而随着维数增加，其性能会迅速下降[1]。文献[2]提出适应性查询选择率的估计，并针对 6 维数据给出实验结果。文献[3]提出一种基于查询集合构造直方图的方法，其实验数据维数为 2~4。文献[4]给出的直方图方法运用 2 维和 3 维人造数据进行实验。对于低维数据，很多估计选择率的方法具有很高精度[1-4]。但对高维数据空间进行有效划分仍是个难题[3]。

本文提出一种新的构造直方图的方法，即基于数据区域的局部分布密度建立直方图，进而用其估计查询选择率。该方法对低维和高维数据皆有效。

2 术语和基本算法

设 $\Re^n$ 为 n 维实数空间，$R \subset \Re^n$ 是一个关系(或数据集)，具有 n 个属性$(A_1, A_2,\cdots, A_n)$，闭区间$[\alpha_i, \beta_i]$是属性 A_i 的值域，D 为关系 R 的值域，即 $D=\prod_{i=1}^{n}[\alpha_i,\beta_i]$。对于$[a_i,b_i]\subseteq[\alpha_i,\beta_i]$，有 $G=\prod_{i=1}^{n}[a_i,b_i]$为一个 n 维超矩形，简称 n-矩形，其体积为 $v(G)=\prod_{i=1}^{n}(b_i-a_i)$。在 2 维空间，$v(G)$表示 G 的面积。

一个区域查询就是检索出数据库中属于 n-矩形 G 的所有元组，即如下形式的查询：

select * from R where $(a_1\leqslant A_1\leqslant b_1)$ and$\cdots$and$(a_n\leqslant A_n\leqslant b_n)$

因此，区域查询 Q 和 n-矩形是对应的。下文不区分术语“区域查询”、“查询”、“区域”和“n-矩形”。

一个直方图 H 是形如 $H=\{(B_1, f_1), (B_2, f_2),\cdots, (B_m, f_m)\}$的集合，$(B_i, f_i)$称为一个桶，$B_i$ 为一个 n-矩形，f_i 称为频率，即 B_i 所包含元组的数量(也记为$|B_i|$)。设 B 为一个桶，用 ρ 表示 B 中元组的分布密度，即 $\rho=f(B)/v(B)$。本文基于区域中元组的分布密度构建直方图，运用文献[5]中的 PPS(Partition of Points in n-dimensional Space)算法和 DTR(Difference of Two Regions)算法，并对 PPS 算法做如下修改：

设 $\Re^n$ 中 m 个点皆属于区域 $G=\prod_{i=1}^{n}[a_i,b_i]$。为了判断 G 的分布是否均匀，用 PPS 算法划分 G 为 p 个小区域$\{P_i: i=1,2,\cdots,p\}$，称 P_i 为一个网格，$|P_i|$(P_i 包含的元组数)最大者称为最大网格 $P_{\max}$，$|P_i|$最小者称为最小网格 $P_{\min}$。算法步骤如下($n_i>1$, $i=1,2,\cdots,k$)：

(1)找出 G 的 k 条最长的边 $e_1,e_2,\cdots,e_k$，把每条边分别分成 $n_1,n_2,\cdots,n_k$ 等份，则 G 被分成 $h=n_1n_2\cdots n_k$ 个小区域$\{P_i: i=1,2,\cdots,h\}$；

(2)找出在$\{P_i: i=1,2,\cdots,h\}$中最大和最小的 2 个网格 $P_{\max}$ 和 $P_{\min}$，若$(|P_{\max}|-|P_{\min}|)\leqslant m_0\times|P_{\max}|$($m_0$ 为一个阈值，用训练方

基金项目：国家自然科学基金资助项目(30971693)；河北大学博士基金资助项目(2009-160)
作者简介：朱 亮(1964－)，男，副教授、博士，主研方向：查询处理和优化，信息检索，人工智能；冯彦超，硕士研究生；刘椿年，教授、博士、博士生导师；杨文柱，副教授、博士研究生
收稿日期：2009-11-07 **E-mail：**zhu@hbu.edu.cn

法得到，在本文实验中，m_0取值为 1/15)，则转(5)；

(3)计算$\{P_i: i=1,2,\cdots,h\}$中每个 P_i对应的 ρ_i，若有$(abs(\rho_j-\rho_i)\leqslant m_0\times\rho_i)and(abs(\rho_j-\rho_i)\leqslant m_0\times\rho_j)$ $(i\neq j)$则将 P_i与 P_j合并，得到$\{P_i: i=1,2,\cdots,k(k\leqslant h)\}$($\rho_i$是 P_i中元组的分布密度)；

(4)$G:=P_i$并转(1)；

(5)结束。

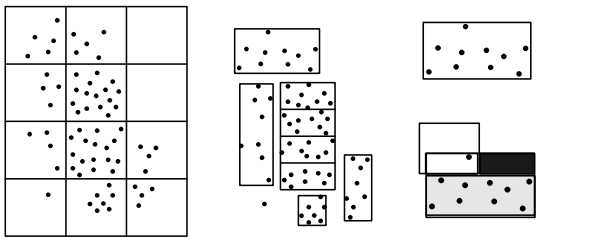
DTR 算法用来计算 2 个 n-矩形之差，设 $S=\prod_{i=1}^{n}[a_i,b_i]$和 $T=\prod_{i=1}^{n}[c_i,d_i]$为 2 个区域。若 $S\cap T\neq\varnothing$，则 $T-S$ 是 T 的一些 n-维子区域的并集，$T-S=\cup_{j=1}^{p}T[j]$，其中，$T[j]\subset T$, $j=1,2,\cdots,p$,$v(T[i]\cap T[j])=0(i\neq j)$。

对于关系 R，若其值域 $D=\prod_{i=1}^{k}[\alpha_i,\beta_i]$中的元组均匀分布，则对于一个区域查询 Q 无需任何算法就可以准确估计其选择率为 $est(Q)=\rho\times v(Q\cap D)=|D|\times v(Q\cap D)/v(D)$，其中，$\rho$ 是 D 的分布密度。鉴于此，对于一个直方图 $H=\{(B_1, f_1), (B_2, f_2),\cdots,(B_m, f_m)\}$，若其中每个桶具有均匀分布，则能准确估计一个查询选择率。因此，基于每个桶的分布密度，本文给出一种建立直方图的新方法，步骤如下：(1)运用 PPS 算法划分一个区域 G，进而判断其是否均匀；(2)合并等密度的网格；(3)抛弃无用区域；(4)收缩网格使其变为包含其所有元组的最小区域；(5)修正最后得到的所有最小区域，构成直方图。

3 直方图的建立

关系 R 的值域 D 作为初始 n-矩形 G。对于每个 n-矩形 G 做如下处理：(1)运用 PPS 算法划分 G 并判断其是否均匀，将 G 划分成若干网格(即小的 n-矩形)$\{P_i: i=1,2,\cdots,h\}$，若 G 的分布是均匀的，则本次划分结束。(2)合并网格，对每个网格 P_i $(i=1,2,\cdots,h)$，计算其密度 ρ_i，对于相邻的网格 P_i与 P_j，若两者密度相近，即$(abs(\rho_j-\rho_i)\leqslant m_0\times\rho_i)and(abs(\rho_j-\rho_i)\leqslant m_0\times\rho_j)$，则将网格 P_i 与 P_j 合并为一个区域。(3)抛弃无用网格，无用网格是指不包含任何元组，或包含元组的个数非常小(小于某个阈值τ)的网格，这些网格对于选择率估计的精度不产生影响，或产生的影响非常小。为节约空间开销，忽略这些网格。(4)收缩网格，对每个网格 P_i，得到包含其所有元组的最小区域，不失一般性地，仍记为 P_i，且令 $G:=P_i$，转第(1)步。(5)修正初始直方图(即所有最小区域的集合)，利用查询反馈方法修正上述步骤得到的初始直方图，运用查询训练集合和 DTR 算法对此直方图进行修正，对于直方图中的一个桶，“挖去”与整个桶的密度相差较大的部分，得到最后的直方图 H。

图 1 描述了数据空间的划分。



(a)G被划分后　　(b)执行 PPS 算法后　　(c)执行 DTR 算法后

图 1　数据空间的划分

如图 1 所示，图 1(a)划分 n-矩形 G 之后，经判断此矩形的分布不均匀，经第(2)步和第(3)步处理后，抛弃图 1(a)中的网格 3、网格 6 和网格 10。合并网格 1 和网格 2、网格 4 和网格 7、网格 5 和网格 8、网格 9 和网格 12，再经过第(4)步收缩网格，分别得到区域 1~区域 4，如图 1(b)所示。对区域 3 再次执行 PPS 算法时，可判之为均匀的。对初始直方图进行修正，如图 1(c)所示，通过训练查询 Q 和 DTS 算法，去掉了区域 1 的黑色部分(即区域 1"，其不含元组)，原来大的区域 1 变为区域 1'和小的区域 1(灰色部分)，记区域 1'为区域 6，最后得到直方图为 $H=\{(B_1, f_1), (B_2, f_2),\cdots, (B_6, f_6)\}$。

4 查询选择率的估计

设 $H=\{(B_1, f_1), (B_2, f_2),\cdots, (B_m, f_m)\}$为直方图，对于区域查询 $Q=\prod_{i=1}^{n}[c_i,d_i]$，先计算查询区域 Q 与 B_i 的交集 $Q\cap B_i$，如图 2 阴影部分所示，然后通过下式计算出区域查询的选择率：

$$est(H,Q)=\sum_{B\in H}f(B)\left(\frac{v(Q\cap B)}{v(B)}\right)$$

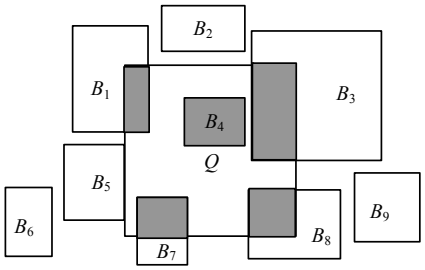


图 2　查询 Q 选择率的估计

特别地，若 B 与 Q 的交集 $B\cap Q$ 仅是一个面(或边)时，如图 2 中 B_5 与 Q 的情况，则用 B 的权重 w_B 进行估计。设 $B=\prod_{i=1}^{n}[a_i,b_i]$,$Q=\prod_{i=1}^{n}[c_i,d_i]$，若 $a_j=d_j$(或 $b_j=c_j$)，则 B 在第 i 维$(i\neq j)$与 Q 可能有边(面)重叠，通过训练方法，得到第 i 维的权重如下：

(1)若 $0<(d_i-a_i)<(b_i-c_i)$, $[a_i,b_i]\not\subset[c_i,d_i]$且$[c_i,d_i]\not\subset[a_i,b_i]$，则 $w_i=(d_i-a_i)/(b_i-a_i)$；

(2)若 $0<(b_i-c_i)<(d_i-a_i)$, $[a_i,b_i]\not\subset[c_i,d_i]$且$[c_i,d_i]\not\subset[a_i,b_i]$，则 $w_i=(b_i-c_i)/(b_i-a_i)$；

(3)若$[a_i,b_i]\subseteq[c_i,d_i]$，则 $w_i=1$；

(4)若$[c_i,d_i]\subset[a_i,b_i]$，则 $w_i=(d_i-c_i)/(b_i-a_i)$。

本文取 $w_B=\prod_{i=1(i\neq j)}^{n}w_i$，并估计 $Q\cap B$ 的元组数为 $est(B,Q)=w_B(f(B)/v(B))$。

5 实验

本文用 Microsoft's SQL Server 2000, VC++6.0 和 Windows XP，及一台配置为 2.8 GHz CPU 和 1 GB 内存的 PC 机。

实验数据集包括低维(2 维~4 维)和高维(25 维和 104 维)实际数据集[3,5]。低维数据集为 Census2D 和 Census3D (皆包含 210 138 个元组)以及 Cover4D(含 581 010 个元组)。高维数据集为 Lsi25D 和 Lsi104D(皆包含 20 000 个元组)。对每个低维数据集，区域查询的集合包含 1 000 个查询，每个查询包含相应数据集所含元组数量的 1%。

对于高维数据集，区域查询的集合生成方法如下：从每个数据集中随机选出 1 000 个元组作为 n-矩形的中心点。对每个属性，边长皆为 $2l$。对 Lsi25D 有 $5.3\times10^{20}<l<3.2\times10^{38}$，对 Lsi104D 有 $6.3\times10^{33}<l<3.5\times10^{38}$，其中，$l$ 为随机实数。

每个训练查询的集合包含 200 个查询，其生成方法同上。用训练查询集合得到各数据集的相关阈值为 $m_0=1/15$, $\tau=10$。

文献[3]报告了方法 STHoles, GenHist, STGrid, Equi-Depth 和 MHist 在低维数据集上的实验结果。下文运用与文献[3]相同的低维数据集和相同参数的随机测试查询集合，对本文提出的基于分布密度的方法与文献[3]方法进行比较。

(1)空间开销。文献[3]中 STHoles 等方法所建的直方图占用的空间大小是 1 000 Byte。本文方法建立直方图时，能够控制其桶个数的上界，但不能预先设定桶的精确个数，对于低维数据集，实验中本文方法的空间开销接近 1 000 Byte。

(2)选择率估计。使用标准绝对误差[3]，用 1 000 个查询度量各种方法在不同数据集上的准确率。标准绝对误差是平均绝对误差 $E(R,H,W)$[3]与一致绝对误差 $E_{unif}(R,W)$[3]的比值。

图 3 显示了 6 种方法的标准绝对误差(即 $E(R,H,W)/E_{unif}(R,W)$)，其中，BDD 方法是本文提出的基于分布密度的方法；其余 5 种方法的数据源自文献[3]中的 Figure12(d)。

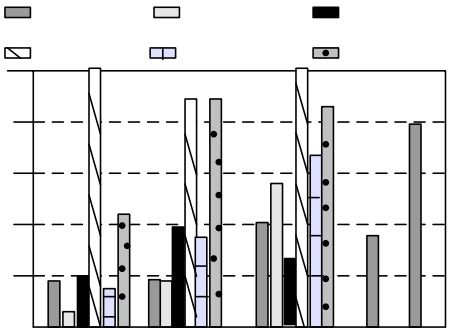


图 3　各种方法所得标准绝对误差

从图 3 可知，对于数据集 Census2D，BDD 方法的性能不如 STHoles 和 Equi-Depth ，但优于其他方法。对 Census3D，BDD 和 STHoles 性能相当，且明显好于其他方法。对 Cover4D，BDD 方法的性能仅次于 GenHist 方法。对于高维数据集 Lsi25D 和 Lsi104D，BDD 方法的标准绝对误差分别为 37%和 79%，其空间开销大于 1 000 Byte，考虑到其直方图中桶的个数分别只有 414 和 330，因此，具有合理的空间开销。由图 3 可知，许多方法的结果大于 40%和 80%，可见，BDD 方法对于高维数据是有效的。

6　结束语

本文方法对低维(2 维~4 维)数据的标准绝对误差为 18%~41%。与现有其他同类方法相比，本文方法具有较高精度。而对高维，即 25 维、104 维数据，该方法也是有效的，其标准绝对误差分别为 37%和 79%。

参考文献

[1] Lee Juhong, Kim D, Chung Chin-Wan. Multi-dimensional Selectivity Estimation Using Compressed Histogram Information[C]//Proc. of 1999 Int'l Conf. on Management of Data. Philadelphia, Pennsylvania, USA: ACM Press, 1999: 205-214.

[2] Lim L, Wang Min, Vitter J S. SASH: A Self-adaptive Histogram Set for Dynamically Changing Workloads[C]//Proc. of the 29th VLDB Conference. Berlin, Germany: [s. n.], 2003: 369-380.

[3] Bruno N, Chaudhuri S, Gravano L. STHoles: A Multidimensional Workload-aware Histogram[C]//Proc. of 2001 Int'l Conf. on Management of Data. Santa Barbara, USA: ACM Press, 2001: 211-222.

[4] He Zhen, Lee Byung-Suk, Wang X S. Proactive and Reactive Multi-dimensional Histogram Maintenance for Selectivity Estimation[J]. Journal of Systems and Software, 2008, 81(3): 414-430.

[5] Zhu Liang, Meng Weiyi, Yang Wenzhu, et al. Region Clustering Based Evaluation of Multiple Top-N Selection Queries[J]. Data and Knowledge Engineering, 2008, 64(2): 439-461.

编辑　陈　晖

(上接第 63 页)

表 1　5 种挖掘算法的规则量和提取率比较

数据集规模	算法	规则量	提取率/(%)
3 000	Apriori	15.0	100
	GA	13.0	86.7
	SAGA	13.2	88.0
	IGA	13.5	90.0
	PCAGS	14.0	93.3
5 000	Apriori	18.0	100
	GA	16.0	88.9
	SAGA	16.0	88.9
	IGA	16.6	92.2
	PCAGS	17.2	95.6
8 000	Apriori	20.0	100
	GA	17.8	89.0
	SAGA	18.8	94.0
	IGA	18.5	92.5
	PCAGS	20.0	100

定义 6　规则量 d 为

$$d=\sum_{i=1}^{n} r_i / n \tag{6}$$

其中，n 表示实验次数；r_i 表示第 i 次实验提取的规则量。

定义 7　提取率 v 为

$$v=\frac{a}{b}\times 100\% \tag{7}$$

其中，b 表示使用经典算法 Apriori 提取的平均规则数；其他算法提取的平均规则数为 a。

6　结束语

传统的 Apriori 算法得到的关联规则不总是相关的、有价值的，有时甚至是误导的，且对大规模数据库而言，该算法执行效率很低，实际可操作性不高。本文在关联规则挖掘中引入规则量和提取率度量机制，提出一种基于并行克隆退火遗传策略的挖掘算法，有效避免了上述问题。

参考文献

[1] 张宗平. 一种更新关联规则的方法[J]. 计算机工程, 2008, 34(1): 64-65, 68.

[2] 赵方方, 刘万军, 陈芳元. 遗传算法在关联规则挖掘中的应用研究[J]. 沈阳理工大学学报, 2006, 25(4): 51-54.

[3] 王礼刚, 左源瑞, 李盛瑜. 一种基于改进型遗传算法的关联规则提取算法及其应用[J]. 重庆师范大学学报: 自然科学版, 2006, 23(2): 42-45.

[4] 武兆慧, 张桂娟, 刘希玉. 基于模拟退火遗传算法的关联规则挖掘[J]. 计算机应用, 2005, 25(5): 1009-1011.

[5] 高　坚. 基于免疫遗传算法的多维关联规则挖掘[J]. 计算机工

程与应用, 2003, 32(2): 185-187.

编辑　陈　晖