一种支持高维数据查询的并行索引机制

王寅峰[1,2] 刘 昊[1] 狄 盛[1] 胡昊宇[1]

(1 香港大学计算机系,香港;2 深圳信息职业技术学院,广东 深圳 518029)

摘要 提出了一种基于独立特征的并行索引体系结构以检索符合正态分布的高维数据.通过分析并行搜索的多维数据及其复杂度,结合用户设定的维度权重返回待核实的结果,最后通过加权相似度计算函数合并检索结果以完成 kNN 查询.针对高维数据的异构性特点,给出了规范情景上下文信息数据的算法.通过联合香港大学的 2 个社区和深圳先进研究院的 CNGrid 社区进行的测试,证明基于并行检索机制的 100NN 查询准确率可达 93%,在千万个高维数据中的检索时间小于 0.7 s,结果表明所提出的并行索引机制能有效提高查询效率,尤其适合海量高维数据的有偏组合特征查询.

关键词 组合查询;高维数据索引;熵;位置敏感哈希函数;中国国家网格

中图分类号 TP393 **文献标志码** A **文章编号** 1671-4512(2011)S1-0156-05

A parallel index mechanism for large scale high dimensional data

Wang Yinfeng[1,2] *Liu Hao*[1] *Di Sheng*[1] *Hu Haoyu*[1]

(1 Department of Computer Science, University of Hong Kong, Hong Kong, China;
2 Shenzhen Institute of Information Technology, Shenzhen 518029, Guangdong China)

Abstract A new parallel index mechanism suitable for searching large scale high dimensional information, with independent features according to normal distribution was proposed. By analyzing parallel search of the multi-dimensional data as well as its combinatorial complexity, a weighted similarity function on user-defined weights to finish the kNN queries was leveraged. In addition, a data normalization algorithm for enhancing the index classification capability was further developed. The methods are evaluated on the CNGrid environment at the University of Hong Kong and Shenzhen SIAT. The 100NN query accuracy is greater than 93% and the response time is less than 0.7 s when searching 10 milllion feature points of images and context information. Theoretical and practical results both indicate that this kind of parallel index mechanism can significantly improve the performance of ad-hoc query composition over large scale high dimensional data in cloud system.

Key words query composition; high dimensional indexing; entropy; locality sensitive hashing; CNGrid

各种类型的交易、Web 文档、用户评分、多媒体等数据日益膨胀,异构(数据模型)与高维的特征(即它们的维度(属性))通常可以达到成百上千维,甚至更高,导致各种应用中需要检索的数据极为复杂且数据量急剧膨胀.同时数据检索越来越多地呈现出有偏查询的特点,即用户基于自身的偏好与在环境交互中的体验,往往仅对数据属性中的某些特征维比较关心,通常选取部分特征作为检索项[1].

与传统高维数据近似查询不同,高维数据有偏组合查询将"权重"引入查询特征中,即每个查询的特征维将具有一定的比例关系以进行 k 近

收稿日期 2011-02-28.

作者简介 王寅峰(1974-),男,博士,E-mail: wangyinfeng@gmail.com.

基金项目 国家高技术研究发展计划资助项目(2006AA01A111);Hong Kong RGC Grant (HKU7176/06E);香港教资会专用设备建设基金资助项目(SEG HKU09).

邻[2]查询，这些特征差异性的组合反映了特征之间的相互作用，并受到各自权重对比的制约. 如何支持海量高维数据中任意多维的特征检索并保证可扩展性与及时性，是确保普适计算与云计算服务质量的一个主要问题.

B 树、R 树、X 树和 M 树[3]等采用基于向量或度量空间[4]划分的方法能够解决低维数据中的索引问题. 但是这些索引机制所需划分的空间随着维数的增加而呈指数增长，导致其查找性能随维数增加急剧下降，不能支持高维数据索引，这种现象被称为维数灾难. 采用空间填充曲线[5]的方法面临单位空间的数目随着维度增加而成倍增长的问题，此外在进行 kNN 搜索时要查找的相邻空间数目也呈指数增长. 由于空间数据稀疏的特性，即使付出很高的地址映射计算代价（如采用多条填充曲线），也不能在确定的时间保证完成近邻搜索. 位置敏感哈希函数（LSH）[6]方法在保证较高查询准确性的前提下，可以有效降低查询的时间复杂度. 然而，预先构建完成的索引机制如何支持用户对任意特征属性的动态有偏组合查询是高维数据索引技术所面临的挑战.

动态有偏组合查询本质上是信息的组合，海量信息的组合极大地增加了数据的复杂性，对系统的可扩展性提出了新的要求. 为了在分布式计算环境中更好地确保系统的可扩展性，充分利用高性能计算的资源加快信息索引的构建与查询，是保证查询结果实时性要求的关键.

为了确保动态有偏组合查询结果的数据质量，并且保证系统响应的实时性与可扩展性，本文提出了一种基于独立特征的并行索引体系结构. 针对 LSH 技术中缺少并行索引机制问题，本研究通过在 CNGrid[7]环境中的多个节点上分别构建、使用不同哈希函数的分布式并行 LSH 服务器，以支持海量数据的索引，保证系统的可扩展性. 在 CNGrid 环境中的大规模数据实验表明系统在保证查询准确性的同时能够确保实时响应.

1　并行索引体系结构

由于 LSH 索引建立时需要占用大量内存保存相对关系计算的中间结果，使用 LSH 工具包[8]建立 1×10^6 个图像特征数据索引时最大使用超过 14 GB 内存；因此，系统将利用 CNGrid 平台所具有的强大计算能力，在多个节点上部署支持 C/S 通信的 LSH 并行处理服务器. 通过在各节点上运行不同的哈希函数，对更多数据建立索引，提高可扩展性. 并行索引查询的工作流程如图 1 所示，具体步骤如下：

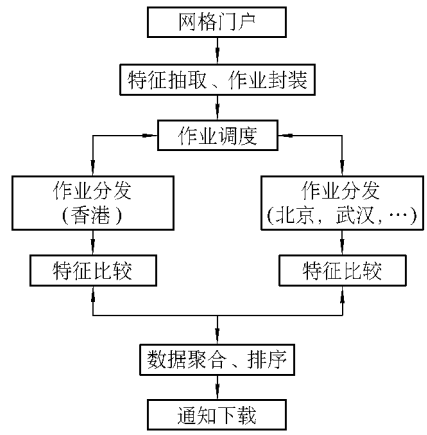

图 1　并行索引系统工作流程

a. 用户在香港发现了比较有趣的瓷器，用手机拍照后上传到香港大学的网格门户进行相似图片搜索，希望得到参考信息；

b. 香港大学网格门户对图片进行特征抽取，并将附加的 context 信息与对图像形状、颜色、纹理特征的偏好，瓷器所在的位置等一并封装为一个请求查询作业，将该作业提交 CNGrid 处理；

c. CNGrid 调度器将该作业分发到香港、深圳、北京等网格节点，调度的作业通过 LSH 服务器进行查询，返回本地数据库中与指定特征最相似的图片，若没有 context 信息，则只在本地网格节点处理；

d. 各网格节点将返回的图片结果和 context 信息通过加权相似度函数进行计算，若图片同时也是地理位置最接近的，则其可能相似的程度更大；

e. 各网格节点的图片与信息返回给香港大学网格节点，香港大学网格节点将聚合的信息合并后按相关度排序，通知用户下载，选择图片后可进一步通过附带的链接了解更多的信息.

2　理论分析

2.1　数据复杂度

对于单一属性的数据集合，其中个体的数目 N 是一定的，个体值相同的越少表明这个集合内部的不同状态越多，即复杂程度越高. 用信息熵[9]公式 $H=-\sum_{i=1}^{k}p_i\log\ p_i$ 表示一系列抽样的复杂性（不确定性）. 数据集合整体的复杂性 $C=NH$. 根据信息熵的定义，不确定性与具体值的大小和单位无关.

当 M 个特征维度内部有 n_1，n_2，…，n_m 个不

同的取值时,kNN 组合查询时返回的数据需要反映这种组合复杂度的不确定性.此外,独立特征索引返回的数据量与组合后向量的整体维度之间存在反比关系,维度越少的索引需返回的数量越多,因为其在整体中起的影响相应较小.kNN 组合查询时每个索引需返回的查询结果为

$$k(H_1+H_2+\cdots+H_m)=k\log(n_1+n_2+\cdots+n_m)\sum_{i=1}^{M}\sum D_i\Big/D_i, \quad (1)$$

式中 D_i 为该索引所对应的维数.在所有维度查询结果返回后,根据用户给出的权重 W_i 加权比较返回最邻近节点的数据.相似度

$$S(N,P)=\sum_{i=1}^{n}D(N_i,P_i)W_i\Big/\sum_{i=1}^{n}W_i, \quad (2)$$

式中的距离函数 D 可以采用欧拉距离、海明距离等.

2.2 数据规范化

以图像搜索为例,针对不同的图片特征提取算法,其数据格式与取值范围存在很大的不同(见表1).若把某张图片的所有特征向量简单地连接起来,并建立 LSH 索引,则会导致某些特征的影响被放大(如 ScalableColor)而某些特征的影响被忽略(如 Correlation)的情形.

欧氏距离的数值大小往往依赖于数值分布范围较大的数据属性,当数据中某几维属性的数值的取值范围相比其他属性过大时,计算出的欧式距离通常只反映了取值较大的属性上的差异性.但在现实中,数值取值较小的属性很可能对分类具有至关重要的作用,因此对于基于欧式距离的相似度函数,数据的规范化至关重要.由于高维数据的异构性特点,因此 context 数值的取值范围可能相差很大.为避免计算相似性距离时忽视取值较小的特征,对 context 信息进行规范化的算法如下:

表1 图像特征取值示例

特征算法	维度	数值实例
Correlation	256	4, 0.291 400 58, 0.279 503 88, 0.271 790 36, 0.266 059 3, 1.0, 1.0 …
FCTH	192	2, 5, 4, 0, 0, 1, 3, 0, 3, 2, 0, 3, 0, 0, 2, 0, 0 …
CEDD	144	0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 …
EdgeHistogram	80	5, 3, 6, 3, 5, 2, 3, 5, 6, 6, 5, 2, 7, 6, 5, 3, 3 …
ScalableColor	64	−119, 41, −39, 58, −14, −4, 16, 27, −31, −13 …
Gabor	60	8.501 860 853 231 33, 0.091 076 306 087 697 97, 8.504 718 696 003 7 …
Tamura	18	3.419 921 875, 5.718 490 256 036 659, 270.0, 256.0, 262.0, 296.0, 243.0 …

Input: Context Feature $i\{a_{ij}\mid a_{ij}\leqslant a_{i(j+1)}, 1\leqslant j\leqslant n\}$; Weight w_i for Feature i: $\{w_i\mid \sum w_i=1\}$

Output: Postprocessing Context Feature $i\{c_{ij}\mid 1\leqslant j\leqslant n\}$

/* Quatitize, Normalize an Weight Context Feature */

Begin:

for (int $i=0$; $i<$ # of features; $i++$) { // Each Context Feature $b_{i*}\leftarrow$Quantize a_{i*}

While preserving relative distance in Context

Max_Value=MAX(b_{i*});

for (int $j=0$; $j<$ # of dimensions; $j++$){ // Each Entry of One Context Feature

$c_{ij}=b_{ij}$/Max_Value; // Normalize

$c_{ij}=c_{ij}w_i$; // Weighted

}

}

End.

例如:在 BetterLife[10] 应用中,天气特征($w=0.2$)取值包括 Sunny, Cloudy, Rainy, Foggy,见表2.

表2 数据规范化示例

特征	量化	归一化	引入权重 $w=0.2$ 后
Sunny	0	0.00	0.000
Cloudy	1	0.33	0.066
Rainy	2	0.67	0.134
Foggy	3	1.00	0.200

3 系统测试与分析

3.1 测试环境与测试用例

测试评价指标为响应时间、准确度与构建索引的时间.响应时间为从用户发出查询请求到收到结果之间的时间.构建索引的时间为从开始读入特征信息到可以开始查询服务的时间.准确度为索引返回结果数量与顺序比较结果数量的比值.相似度由式(2)计算.

图像数据为对 9.2×10^5 张图片(数据来源 http://press.liacs.nl/mirflickr/dlform.php)进行特征抽取后的真实图像特征数据,对每种图像

特征分别选取 1.0×10^5，5.0×10^5 和 9.2×10^5 的数据规模进行测试，测试参数见表 3.

表 3　测试参数

取值范围	特征维度	图像数据量
大	>100	$>6\times10^5$
中	(50,100]	$(1\times10^4, 6\times10^5]$
小	[20, 50]	$[0, 1\times10^4]$

LSH 是通过类似超球覆盖的方式查找相邻的点. 根据不同的特征数据分布与取值范围，需要采用不同的查询半径以建立并行 LSH 索引. 通过式(1)计算出最优参数，可以避免某个特征索引返回过多的结果，降低数据本身的差异对实验结果的影响. 测试中索引的查询半径 R 见表 4.

表 4　测试系统所参考的查询半径

特征编号	特征名称	特征维度	查询半径
1	Correlation	256	2.0
2	FCTH	192	8.0
3	EDGE	80	19.0
4	ScalableColor	64	43.0
5	Gabor	60	1.2
6	Tamura	18	135.0

系统测试环境为香港大学 16 个网格节点，每个节点 16 个 CPU，内存 16 GB. 深圳 10 个网格节点，每个节点 16 个 CPU，内存 32 GB. 所有节点均为 64 位 GNU/Linux 系统.

3.2　测试结果

3.2.1　查询时间

对不同维度的 6 种图像特征，在不同数据规模下查询，并记录查询完成所需的时间 t_c.

从图 2 可知对于索引由 256 维图像特征描述的 9.2×10^5 个图像进行查询，平均查询时间在

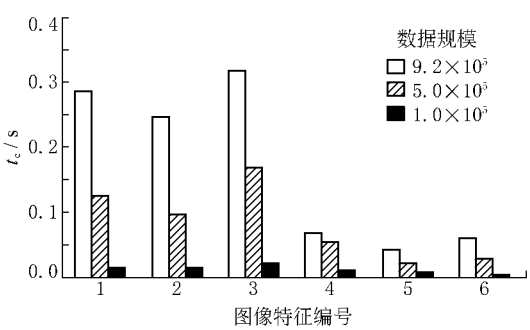

图 2　数据规模与平均查询时间

0.3 s 以内，能够满足实时应用的需要. EDGE 和 Tamura 在与其他图像特征进行比较时，响应时间略有不同. 这是由于 LSH 的理论基础是建立在数据分布为正态分布的假设之上的，而真实图像中 2 个图像特征值的数据分布与正态分布存在差异，因而增加了查询时间.

每个 LSH 服务器建立索引所需的时间如图 3 所示，索引建立时间 t_j 在 200 s 左右.

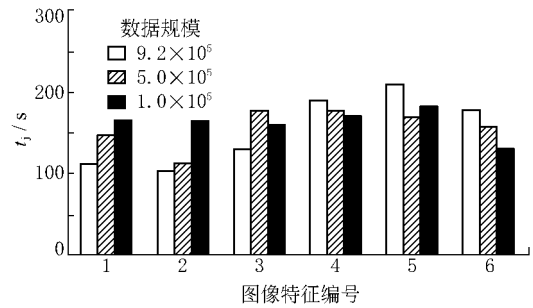

图 3　数据规模与建立索引所需时间

3.2.2　查询准确率

测试中将不同维度的图片特征在不同的数据规模下，对每个特征查询 100 次. 每次随机抽取 1 幅图片的特征进行 100NN 的检索，然后将结果与完全查找的结果进行对比，以评价并行索引查询的准确率. 在每个 LSH 服务器建立索引前，其准确率参数预设为 0.899 999 976，从图 4 中可以看出各项测试的准确率均高于 93%.

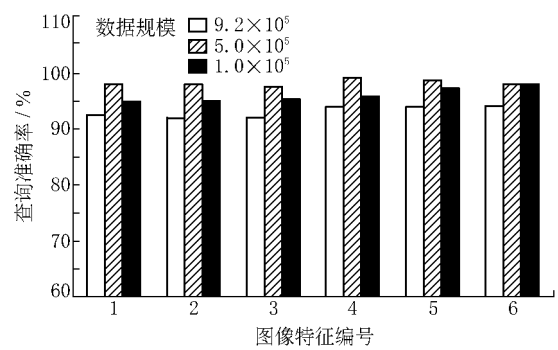

图 4　数据规模与查询准确率

3.2.3　查询半径测试

该测试采用图像特征 Tamura，并对其进行多半径查询，用来测试查询半径对查询结果的正确性以及对查询时间的影响. 缺省 LSH 索引准确率参数设置为 0.899 999 976，如图 5 所示，可

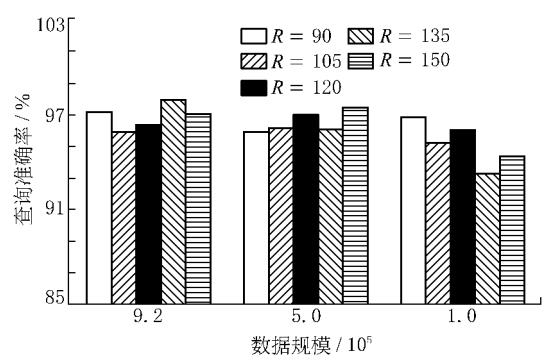

图 5　查询半径与准确率

以发现准确率不受查询半径的影响. 在不同的测试数据规模和查询半径下，其准确率都大于预设的准确率参数.

由图6(t_p 为平均查询时间)可知,查询半径对于并行 LSH 索引的性能会产生一定的影响,在数据量相同的情况下,查询时间随着查询半径的加大而增加.

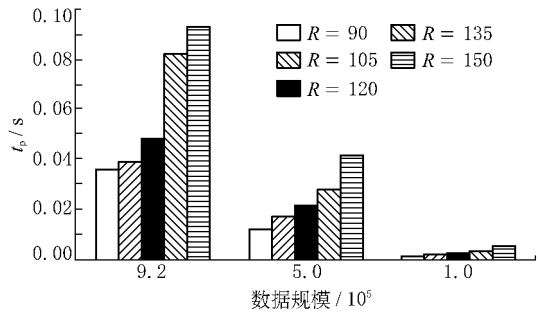

图6 查询半径与平均查询时间

3.3 广域环境中的测试

对部署在深圳先进技术研究院的 1×10^7 规模的高维图像特征数据进行并行查询,数据文件的每一行为一个 SIFT 高维数据的信息,数据规模见表5.

表5 建立并行索引所使用的数据文件规模

名称	文件规模/B
sift-1.dts	1 604 600 214
sift-2.dts	1 587 107 770
sift-3.dts	1 565 022 434
sift-4.dts	1 573 817 598
sift-5.dts	1 556 400 549
sift-6.dts	1 554 420 071
sift-7.dts	1 583 541 427
sift-8.dts	1 548 327 547
sift-9.dts	1 557 634 868
sift-10.dts	1 544 197 122

并行索引服务器分别建立于深圳先进技术研究院 Dawning5000A 机群的 node{8, 30, 31, 33, 34, 36, 37, 39, 41, 42}上.测试时通过 CNGrid 的 portal(http://210.75.252.55:8080/hpc-gapp/)登录后随机选取图片进行 100 NN 的相似图片检索,重复进行 50 次检索请求,记录系统返回最终结果的响应时间 t_x 如图7所示.

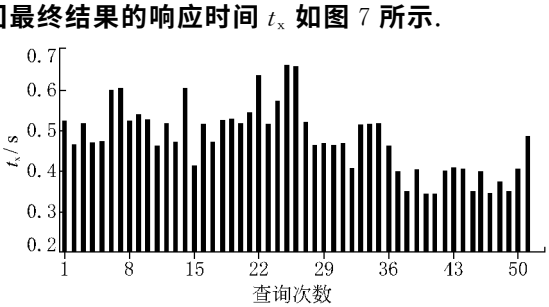

图7 并行查询响应时间

从图7中可以看出:通过 CNGrid 对千万级规模的高维数据进行并行查询处理响应时间在0.3~0.7 s之间,可以满足用户对实时性的要求.

本文针对海量高维数据索引中用户有偏特征组合查询的要求,提出了一种适用于云计算模式的并行索引机制.在查询中通过对组合查询数据复杂度的分析确定每个索引需要返回的结果数目,并根据加权相似度函数计算最终结果.在下一步的工作中,准备结合智能推理的方法提高相似性搜索的准确率,并在保证索引服务器性能的前提下降低索引的空间代价,以支持索引更多的高维数据.

参 考 文 献

[1] de Mantaras R L, McSherry D, Bridge D, et al. Retrieval, reuse, revise, and retention in CBR[J]. The Knowledge Engineering Review, 2006, 20(3): 215-240.

[2] Yiu M L, Mamoulis N. Reverse nearest neighbors search in ad-hoc subspaces[J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(3): 412-426.

[3] Ciaccia P, Patella M, Zezula P. M-tree: an efficient access method for similarity search in metric spaces [C]// 23rd VLDB Conference. Athens: VLDB, 1997: 426-435.

[4] Zezula P. Similarity search the metric space approach [M]. Berlin: Springer, 2006.

[5] 高迎,程涛远,王珊. 基于 Hilbert 曲线的许可证存储策略及查找算法[J]. 软件学报, 2006, 17(2): 305-314.

[6] Andoni A, Indyk P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions [J]. Communications of the ACM, 2008, 51(1): 117-122.

[7] 中国国家网格运行管理中心. 中国国家网格. [2010-10-10]. http://www.cngrid.org.

[8] Andoni A. LSH toolkit. [2010-09-28]. http://web.mit.edu/andoni.

[9] Shannon, Claude E. Prediction and entropy of printed English[J]. The Bell System Technical Journal, 1951, 30: 50-64.

[10] Hu D, Wang Y F, Wang C L. BetterLife 2.0: large-scale social intelligence reasoning on cloud [C]// IEEE CloudCom 2010. Indianapolis: IEEE, 2010:529-536.