基于 LSH 索引的快速图像检索

唐俊华　阎保平
（中国科学院网络中心数据库室，北京 100080）
E-mail：tangjunhua@sdb.ac.cn

摘　要　高维空间中点数据的索引及检索是基于内容图像检索领域的关键问题，文中将 LSH（locality sensitive hashing）索引算法应用于基于内容图像检索系统中，与传统的索引方法相比，该算法具有复杂度比较低、支持非常高的维数、极低的 I/O 代价等特点。实验结果证明，将该索引算法应用于基于内容图像检索系统中，其性能优于传统的索引方法.

关键词　基于内容图像检索　索引结构　相似性检索　LSH 算法

文章编号 1002-8331-（2002）24-0020-02　**文献标识码** A　**中图分类号** TP391

Fast Image Retrieval Based on LSH Indexing

Tang Junhua　Yan Baoping
（Database Department，Computer Network Information Center，The Chinese Academy of Sciences，Beijing 100080）

Abstract：It is a critical issue for indexing and retrieval of high dimensional point data in content-based image retrieval field.In this paper，a new kind of indexing structure is adopted in the content-based image retrieval system，in comparison with traditional indexing methods，the LSH can build with low complexity，support very high dimensionality，and even very low I/O cost，etc.

Keywords：Content-based image retrieval，Index structure，Similarity retrieval，LSH algorithm

1　引言

在基于内容的图像检索系统中，需要在图像集中查找与某个给定图像“相似”的图像，这样的查找过程叫“相似性检索”。通常，是从图像中提取“特征”，然后在图像的“特征”上定义相似性。大多数情况下，特征用多维空间的点（或矢量）来描述，空间的维数可能低至几维，也可能大至数千维。空间中特征矢量之间的接近程度反映了对象内容的相似程度，因此基于内容的检索就简化为空间中点的快速搜索问题。

实现相似搜索的方法很多，但是存在一种最基本的方法，叫做“顺序扫描算法”（Sequential Scan Algorithm，SSA）：顺序检查对象集合中的每个对象是否符合相似性检索要求。SSA 的开销很大。多年来，人们已经开发出多种支持相似性检索的索引结构，例如 R-树[1]，K-D-树[2]，K-D-B-树[3]，SS-树[4]和 SR-树[5]等，这些方法都可以归类为空间划分的方法，在文献[6]中指出，随着维数的增加这一类的索引方法的检索性能会急剧恶化。例如，当维数增加时，R-树的检索时间复杂度将很快的接近 $O(k)$。

在很多情况下，做相似图像检索时，快速地检索出一个大致符合要求的图像集合，往往比花费很长的时间检索出与检索要求完全相符合的图像集合更具有吸引力。这一点在数据规模较大，并且对响应时间有较高要求的场合显得尤为重要。LSH（Locality Sensitive Hashing）算法首先由 Indyk 和 Motwani[7]提出，用来解决主存储器中的最近邻居搜索问题。该文将 LSH（Locality Sensitive Hashing）应用于基于内容的图像检索。实验证明，在数据规模、维数增大时，该索引方法仍然具有很好的性能。

2　问题定义

为了方便说明，首先定义几个记号。用 O 表示图像集合，映射 $f:O\to X$ 将 O 中的每幅图像映射到 N 维特征矢量数据集 X 中的一个矢量（也就是 R^N 中的一个点）上。假设采用的距离测度为 $D:R^N\times R^N\to R^+\cup\{0\}$，其中 R^+表示正实数空间，则相似度可以由 $D(\cdot,\cdot)$衡量，$D(\cdot,\cdot)$越小，相似度越大。

如果用 $R^Q(x_0)$表示查询结果集，其中 x_0 表示查询对象，传统的“k-近邻查询（k-nearest neighboring query）”（k 为给定的一个正整数），可以定义如下：

当 $k=1$ 时，$R^Q_{1-nearest}(x_0)=\{x|x\in X,\forall y\in X:D(x,x_0)\geq D(x,x_0)\}$，设 $R^Q_{(k-1)-nearest}(x_0)$为“$(k-1)$最近邻查询”的结果，则“k-近邻查询”定义为：

$$R^Q_{k-nearest}(x_0)=R^Q_{(k-1)-nearest}(x_0)\cup\{x|x\in X',\forall y\in X:D(x,x_0)\geq D(x,x_0)\}$$

其中 $X=X-R^Q_{(k-1)-nearest}(x_0)$。

在上一节中提出，快速地检索出一个大致符合要求的图像集合，往往比花费很长的时间检索出与检索要求完全相符合的图像集合更具有吸引力。基于这个想法，定义“k-近邻查询”的基础上定义“近似 r-近邻”。

定义 1：（k-近邻查询）对于 N 维特征矢量数据集 X，以及

基金项目：国家 863 高科技发展计划资助项目　编号：863-306-ZD11-03-3）

作者简介：唐俊华（974~），男，四川绵阳人，博士生，主要研究领域为基于内容图像检索，分布式信息系统。阎保平（950~），女，山东青岛人，博士，研究员，博士生导师，主要研究领域计算机网络，大型信息系统。

一个给定的正实数 r，在测度 $D(\cdot,\cdot)$ 下满足条件 $\{x|D(q,x)\leq r\}$ 的点集 P，称 P 为 q 的 r–近邻。

近似“r–近邻查询”定义如下：

定义 2：近似 r–近邻）对于一点集 $P\subseteq X$，若对于其中的任一 $x\in P$，有 $P(D(q,x)\leq r)\geq p_1$，即 P 中的数据点与查询 q 的距离小于 r 的概率大于 p_1，则称 P 为 q 的近似 r–近邻。

运用 LSH 算法可以快速地获得查询请求 q 的“近似 r–近邻”，返回该数据点集中的 k 个与查询请求 q 最相似的点，就可以解决 k–近邻查询问题。

3 LSH 算法描述

LSH 算法的基本思想是对数据点集，利用一组具有一定约束条件的哈希函数来建立多个哈希表，使得在某种相似度量条件下，相似的点发生冲突的概率较大，而不相似的点发生冲突的概率相对较小。在文中做如下的假设：

（1）使用 L_1 距离作为相似性度量，对于两个数据点 p,q，L_1 距离被定义为 $\sum_{i=1}^{N}|p_i-q_i|$；

（2）每一个数据点的每一维都是正整数。

对于假设（1），在大多数基于内容图像检索系统中都采用 L_1 距离，或者 L_2 距离作为相似度量，两者在检索性能上没有明显的差异。对于假设（2），在一定的精度要求下，选取一个足够大的正数，与每一个数据点做乘积运算即可保证该假设成立。

首先引入 LSH 函数的定义：

定义 3：一组哈希函数 $H=\{h_1,\cdots,h_k\}$，对于数据点 p,q，若 $D(p,q)<r_1$，则 $P[h_i(p)=h_i(q)]>p_1$，若 $D(p,q)>r_2$，则 $P[h_i(p)=h_i(q)]<p_2$，其中函数 $P(\cdot)$ 是概率函数，i 为随机数，$i\in\{1,\cdots,k\}$，这组哈希函数被称为以 (r_1,r_2,p_1,p_2) 为参数的 LSH 函数组。

对于 d 维的海明空间（即 d 维的二进制串空间）中的点数据 $p=\{p_1,\cdots,P_d\}$，$q=\{q_1,\cdots,q_d\}$，其海明距离被定义为不相同位的个数，令 $r_1=r$，$r_2=(1+\varepsilon)r$，则哈希函数组 $H=\{h_1,\cdots,h_d\}$，$h_i(p)=p_i$，是一个以 $(r,(1+\varepsilon)r,1-\frac{r}{d},1-\frac{(1+\varepsilon)r}{d})$ 为参数的 LSH 函数组。进一步的，构造一组包含个哈希函数的函数组 $G=\{g_1,\cdots,g_k\}$，其中 $g_i(p)=(h_{i_1}(p),h_{i_2}(p),\cdots,h_{i_l}(p))$（即 $g_i(p)$ 为从二进制串 p 中随机选取 l 个不重复位所形成的新的二进制串），$i_1,\cdots,i_l$ 是从集合 $\{1,\cdots,d\}$ 中随机选取的 l 个不重复的整数，容易证明函数组 G 也是一个以 $(r,(1+\varepsilon)r,P_1,P_2)$ 为参数的 LSH 函数组，并且有下面的定理成立。

定理 1：对于具有 N 个 d 维二进制串数据点集，给定 $r>0$，$\varepsilon>0$，若使得哈希函数的个数 $k=\log_{\frac{1}{p_2}}N$，每一个哈希函数所取的随机位数 $l=N^{\lambda}$，取 $\lambda=\frac{\ln\frac{1}{p_1}}{\ln\frac{1}{p_2}}$，$p_1=\frac{r}{d}$，$p_2=1-\frac{(1+\varepsilon)r}{d}$，则 $P_1\geq 1-\frac{1}{e}$，$P_2<\frac{1}{2}$，并且对于该索引结构的检索算法的时间复杂度为 $O(N^{\frac{1}{1+\varepsilon}})$。该定理的详细证明可参考文献[7]。

回到第 2 节中的问题定义，在基于内容的图像检索系统中，图像集合 O 通过映射 $f:O\to X$ 将 O 中的每幅图像映射到 N 维特征矢量数据集 X 中的一个矢量（也就是 R^N 中的一个点）上，对于其中的每一个矢量 $x\in X$，可以方便地将其转化为海明空间上的点数据，方法如下：

假设 C 为一足够大的正整数，与 X 中的每一个向量相乘，使得矢量数据集 X 满足假设 2。设在数据集 X 中，所有维中最大值为 K，对于给定的 R^N 空间中的点数据 $x_1=(x_{11},\cdots,x_{1N})\in X$，可以将其转化为一个 $K\cdot N$ 维的二进制串 $v(x_1)=U(x_{11})U(x_{12})\cdots U(x_{1N})$，其中 $U(x)$ 为 x 个“1”，与 $K-x$ 个“0”连接所构成的二进制串，很明显两个点数据之间的 L_1 距离与其对应的两个 $K\cdot N$ 维的二进制串之间的海明距离是相等的。

这样，对于数据点集 X，索引算法可以描述为下面三个步骤：

算法 1：

（1）将数据点集转化为海明空间中的二进制串。

（2）选取合适的 $r>0$，$\varepsilon>0$，随机选取 k 个上文所述形如 $g_i(p)=(h_{i_1}(p),h_{i_2}(p),\cdots,h_{i_l}(p))$ 的哈希函数。

（3）利用这些哈希函数，将数据点存入相应的哈希表项。

检索算法可描述如下：

算法 2：

（1）对于查询 q，运用算法 1 中的 k 个哈希函数，提取 $g_i(q)$，$1\leq i\leq k$ 所击中的哈希表项。

（2）对这些表项顺序排序，即得到检索结果。

需要指出的是在实际计算过程中，不必真正将数据点转化为二进制串，而只需计算出哈希函数值即可。

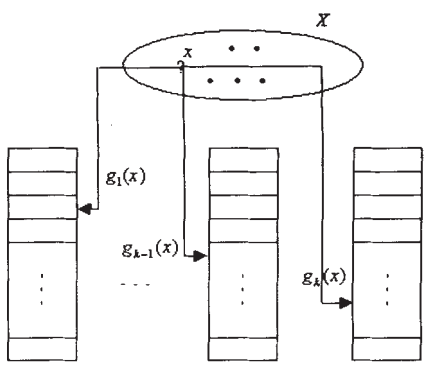

图 1 LSH 索引结构图

如此，数据点的索引以及查询就转化为一组哈希函数操作，以及在一个相对很小的数据集合上的顺序检索。根据该节前一部分的讨论可知，通过哈希函数所获得的数据点集能保证大部分的点数据符合检索要求。参数可以作为计算复杂性与检索精度的折衷因子。可以对不同的值建立多个索引结构，以满足对不同距离查询的要求，在大多数应用情况下，有一个索引就能够满足要求。

4 实验分析

在实验中，文章采用一个 5000 幅图像的图像库，每一幅图像提取 H 维的色彩直方图，该文主要在不同数据规模、不同维数条件下，从检索时间以及误检率来分析该算法的性能，在实验中取，$r=4$，$\varepsilon=2$。文章实现了文献[5]中的 SR–树索引方法，与该文所述算法对照，下面是实验数据及其分析。

图 1 中对 H 取不同值条件下，检索时间的统计平均。图 2 中对 $H=128$，即维数一定条件下，不同数据规模时检索时间的统计平均。

（下转 63 页）

都收敛到了全局最大值，从图形上看至少要经过 20 次迭代才收敛。表 4 列出了 ACT-GA30 次运行的统计结果。ACT-GA 的参数设置如下：群体规模 20；变异概率 0.1；变异切换间隔为 9，即不使用高斯变异；迭代次数 8。从表 4 可知，ACT-GA 仅需 8 次迭代就得到了很满意的结果，可见 ACT-GA 克服过早收敛能力是很强的。

表 4

平均函数计算次数（迭代次数）	函数均值（标准差）	最差优化值	最好优化值
22018（8）	0.999875（5.64×10^{-4}）	0.99819588657	0.99999999997

以上实验表明，整体轮换杂交法对提高遗传算法克服过早收敛能力是很有效的。

5 结论

文章针对遗传算法先选择后杂交的方法存在搜索过于集中，容易造成群体多样性降低，导致过早收敛的问题，提出了整体轮换杂交法。方法的要点是让整个群体的个体尽可能多地先配对杂交，再进行局部竞争选择，从而达到保持群体的多样性，提高算法克服过早收敛能力的目的。数值实验和比较表明整体轮换杂交法可以更有效地避免算法过早收敛。

（收稿日期：2002 年 7 月）

参考文献

1.Francisco Herrera，Manuel Lozano.Gradual Distributed Real-Coded Genetic Algorithms[J].IEEE Trans on Evolutionary Computation，2000；5（）：43~63

2.J Craig Potts，Terri D Giddens，Surya B Yadav.The Development and Evaluation of an Improved Genetic Algorithm Based on Migration and Artificial Selection[J].IEEE Trans on System，Man and Cybernetic，1994；24（）：73~85

3.陈长征，王楠.遗传算法中交叉和变异概率选择的自适应方法及作用机理[J].控制理论与应用，2002；19（）：41~43

4.周明，孙树栋.遗传算法原理及应用[M].北京：国防工业出版社，1996

5.Bäck T.Selective Pressure in Evolutionary Algorithms：A Characterization of Selection Mechanisms[C].In：CEC'94，1994；1：57~62

6.潘正君，康立山，陈毓屏.演化计算[M].北京：清华大学出版社，1998

7.[日]玄光男，程润伟.遗传算法与工程设计[M].北京：科学出版社，2000

8.Z 米凯利维茨.演化程序——遗传算法和数据编码的结合[M].北京：科学出版社，2000

9.林丹，李敏强，寇纪淞.基于实数编码的遗传算法的收敛性研究[J].计算机研究与发展，2000；37（1）：1321~1327

10.Yiu-Wing Leung，Yuping Wang.An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization[J].IEEE trans on Evolutionary Computation，2001；5（）：41~53

（上接 21 页）

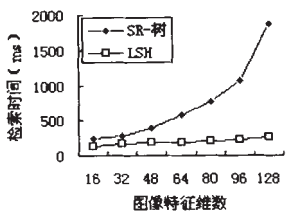

图 2 不同维数的检索时间比较

在不同条件下，笔者统计了该算法的误检率，误检率定义如下：假设利用 LSH 得到包含 K 个点数据集 P_{LSH}，利用“顺序扫描算法”（Sequential Scan Algorithm，SSA）也得到包含 K 个点数据集 P_{SSA}，若点数据 $x\in P_{LSH}$ 且 $x\notin P_{SSA}$，则该数据点为一个误检。对于 N 次检索平均误检率可表示为 $\frac{E}{K\cdot N}$，其中 E 为 N 次检索中误检数据点的总数。在绝大多数情况下都能够保证误检率小于 5%。

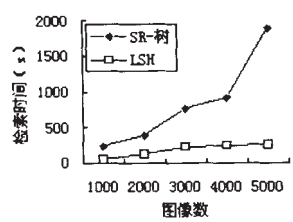

图 3 不同数据规模检索时间比较

通过实验数据，不难看出 LSH 索引算法在数据规模、数据维数的增加时，仍具有良好检索性能。而对照算法在这种情况下，性能会急剧得下降。相对于对照算法，LSH 算法在将误检率控制在一定范围的前提下，能够很大幅度地提高检索性能，将该算法运用于基于内容图象检索系统中是合适的。

5 结论

文章主要讨论了在基于内容图像检索系统中高维点数据的相邻查询问题，将 LSH 索引算法应用于图像检索系统中。实验结果表明，该算法在数据点维数较高的情况下仍然具有良好的检索性能。并能够保证将误检率控制在一定范围。

（收稿日期：2002 年 8 月）

参考文献

1.Guttman A R-trees.A dynamic index structure for spatial searching [C].In：Proceedings of the ACM SIGMOD International Conference on Management of Data，Boston，MA，1984：47~57

2.Bentley J L.Multidimensional binary search trees used for associative searching[J].Communications of the ACM，1975；18（9）：509~517

3.Robinson J T.The K-D-B-tree：A search structure for large multidimensional dynamic indexes[C].In：Proceedings of the ACM SIGMOD International Conference on Management of Data，Michigan，1981：10~18

4.White D A，Jain R.Similarity indexing with the SS-tree[C].In：Proceedings of the 12th International Conference on Data Engineering，New Orleans，LA，1996：516~523

5.Katayama N，Sotoh S.The SR-tree：An index structure for high dimensional nearest neighbor queries[C].In：Proceedings of the ACM SIGMOD International Conference on Management of Data，Tucson，Arizona USA，1997：369~380

6.Weber R，Schek H-J，Blott S.A quantitative analysis and performance study for similarity search methods in high-dimensional spaces [C].In：Proceedings of the 24th VLDB Conference，New York，1998：194~205

7.Indyk P，Motwani R.Approximate nearest neighbor-towards removing the curse of dimensionality[C].In：Proceedings of the 30th Symposium on Theory of Computing，1998：604~613